

ABSTRACT

Title of Dissertation: SELECTED PROBLEMS IN
MANY-REVOLUTION TRAJECTORY
OPTIMIZATION USING Q-LAW

Jackson Shannon
Doctor of Philosophy, 2021

Dissertation Directed by: Professor Christine Hartzell
Department of Aerospace Engineering

Q-Law is a Lyapunov guidance law for low-thrust trajectory design. Most prior implementations of Q-Law were limited to relatively simple low-thrust transfers. This work aims to improve the optimality, usability, and efficiency of Q-Law for better application to the mission design process. To accomplish this, Q-Law is combined with direct collocation to form an efficient hybrid method for high-fidelity, many-revolution trajectory design. Additionally, forward and backward Q-Law propagation are combined to form a novel method for Lunar transfer trajectories. This technique rapidly produces spiral trajectories to the Moon and provides mission designers with a means for efficient trade space exploration. Additionally, backward propagated Q-Law is combined with heritage trajectory design software to produce spiral escape trajectories as well as single and double Lunar swingby trajectories for interplanetary rideshare mission scenarios. Lastly, analytical partial derivatives of the Q-Law thrust vector calculation are derived, and the Q-Law algorithm is wrapped in a nonlinear programming problem. When these derivatives are

used to generate the trajectory state transition matrix, the efficiency and accuracy of the optimization is superior to finite difference solutions. Using this approach, a novel Q-Law multiple shooting method is formulated and tested on various low-thrust transfer problems. These enhancements to the standard Q-Law algorithm enable efficient trade space exploration for more complex low-thrust trajectories, with a specific emphasis on the needs of SmallSat rideshare missions.

SELECTED PROBLEMS IN MANY-REVOLUTION TRAJECTORY OPTIMIZATION USING Q-LAW

by

Jackson Shannon

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Professor Christine Hartzell, Chair/Advisor
Professor Raymond Sedwick
Professor Roberto Celi
Professor Olivier Bauchau
Professor Douglas Hamilton
Dr. Donald Ellison

© Copyright by
Jackson Shannon
2021

Dedication

To Mom, Dad, and Kelly.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and supported me throughout my graduate experience. First and foremost, I would like to thank my advisor, Dr. Christine Hartzell, for her guidance throughout my graduate career and giving me the opportunity to work in her lab. After beginning the graduate program without a clear thesis topic in mind, she gave me the support and flexibility to seek out a research field I am passionate about, and for that, I am very grateful. She is a wonderful person and professor, and it has been a privilege to learn from her. I am also grateful for my lab mates in the Planetary Surfaces and Spacecraft Lab: Thomas Leps, Eric Frizzell, Charles Pett, Anand Patel, and Dr. Alexis Truitt. They helped me prepare for the qualifying and comprehensive exams and provided useful discussions and collaborations on both coursework and research. Additionally, I would like to thank my thesis committee members, Dr. Raymond Sedwick, Dr. Roberto Celi, Dr. Olivier Bauchau, and Dr. Douglas Hamilton, for their time and constructive feedback throughout this process.

I am grateful for the support I have received from both the Johns Hopkins Applied Physics Laboratory (APL) and NASA. The early phases of this research began in the APL summer internship program under the direction of Dr. Martin Ozimek and Dr. Justin Atchison. Without their continued mentorship over the

last few years, the potential of this research would not have been realized. Later on, this research was funded by the NASA Space Technology Research Fellowship through grant # 80NSSC19K1172 and in collaboration with Dr. Donald Ellison. His mentorship was invaluable in completing the final stages of this research. Marty, Justin, and Donald, it has been an honor to work with and learn from each of you.

To my family, thank you so much for your unwavering love and support. Mom and Dad, I would not be where I am today without you. Any successes I may have are a direct product of your love and dedication. Kelly, you are the best sister I could have asked for, and you continue to inspire me everyday. I could not have made it through this program without you guys. Thank you for everything.

To Haley, you have been the biggest source of support throughout this process. You stood by me from the first moment I considered pursuing graduate school and have helped me overcome countless challenges throughout the program. You are the most wonderful person I have ever met, and I could not ask for a better person to spend the rest of my life with. Thank you for always having my back.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	v
List of Tables	vii
List of Figures	viii
Chapter 1: Introduction	1
1.1 Problem Definition and Motivation	1
1.2 Low-Thrust Many-Revolution Design Methods	3
1.3 Dissertation Overview	7
1.4 Summary of Contributions	9
Chapter 2: System Dynamics	11
2.1 Perturbed Orbital Motion	11
2.2 Orbital Element Sets	14
2.2.1 Classical Orbital Elements	16
2.2.2 Modified Equinoctial Elements	18
2.3 Eclipse Model	20
Chapter 3: Q-Law Lyapunov Guidance Algorithm	24
3.1 Lyapunov Control Functions	24
3.2 Q-Law Thrust Vector Calculation	26
3.3 Gain Tuning	30
3.4 Basic Transfers	31
3.5 Fast Variable Targeting	35
Chapter 4: Direct Collocation with a Q-Law Initial Guess	46
4.1 Direct Collocation	48
4.2 Nonlinear Programming	54
4.3 Numerical Differentiation Techniques	56
4.4 Optimization Problem Setup	58
4.5 Example: GTO to GEO Transfer	60
4.5.1 Time-Optimal Results	61

4.5.2	Mass-Optimal Results	70
Chapter 5:	Low-Thrust Lunar Transfers	74
5.1	Forward-Backward Q-Law	76
5.2	Optimization Problem Setup	79
5.3	Example: SMART-1 Mission	81
5.4	Example: GTO-Moon Mission	88
5.5	Example: LEO to LLO Transfer	101
Chapter 6:	Low-Thrust Lunar Swingby Escape Trajectories	106
6.1	Perturbed Sims-Flanagan Transcription	107
6.2	Monotonic Basin Hopping	110
6.3	Lunar Swingby Design	112
6.4	Example: Comet 45P Flyby	116
Chapter 7:	Derivation and Application of Analytical Partial Derivatives of the Q-Law Thrust Vector	124
7.1	Computation of the Q-Law Thrust Vector Partial Derivatives	127
7.1.1	Partial Derivatives With Respect to W_a	128
7.1.2	Partial Derivatives With Respect to W_e	129
7.1.3	Partial Derivatives With Respect to W_i	129
7.1.4	Partial Derivatives With Respect to W_ω	130
7.1.5	Partial Derivatives With Respect to W_Ω	131
7.1.6	Partial Derivatives With Respect to Semi-Major Axis	132
7.1.7	Partial Derivatives With Respect to Eccentricity	133
7.1.8	Partial Derivatives With Respect to Inclination	133
7.1.9	Partial Derivatives With Respect to Argument of Periapsis	134
7.1.10	Partial Derivatives With Respect to Longitude of Ascending Node	135
7.1.11	Partial Derivatives With Respect to True Anomaly	136
7.1.12	Partial Derivatives With Respect to Mass	136
7.2	Shooting Methods	137
7.3	Q-Law Shooting Setup	139
7.4	Logistic Function Coasting	146
7.5	Example: GTO to GEO Transfer	148
7.6	Example: Lunar Transfer	151
7.7	Example: Direct Launch and Capture at Mars	156
7.8	Example: Lunar Swingby Escape to Comet 45P	160
Chapter 8:	Conclusions	164
8.1	Summary	164
8.2	Publications and Presentations	170
8.3	Future Work	171
	Bibliography	174

List of Tables

1.1	Qualitative aspects of many-revolution low-thrust transfer methodologies.	7
3.1	LEO-GEO problem specifications.	32
3.2	Equatorial to polar problem specifications.	34
3.3	Eccentric orbit transfer problem specifications.	39
3.4	Results for iterative \overline{M}_0 targeting.	45
4.1	Initial and target orbits in equinoctial coordinates	66
4.2	Dynamics constants, Q-Law parameters, and GPOPS-II settings.	67
4.3	Time-optimal problem results for varying patch point location and literature comparison	69
4.4	Mass-optimal results for the 50% case and Q-Law with effectivity coasting mechanism. The 50% time-optimal and original Q-Law solutions are also presented for comparison.	72
5.1	SMART-1 Problem Parameters	83
5.2	Trajectory optimization results for the SMART-1 problem and literature comparison.	84
5.3	ESPA-Class Problem specifications.	88
5.4	GTO-Moon Problem MOEA decision vector and bounds.	92
5.5	LEO-LLO Problem MOEA decision vector and bounds.	103
5.6	LEO-LLO Problem specifications.	103
6.1	Comet 45P Flyby Problem specifications.	117
7.1	Initial and target orbits	148
7.2	GTO to GEO transfer dynamics constants and Q-Law parameters.	149
7.3	Minimum-time GTO-GEO problem using NLP wrapped Q-Law.	150
7.4	Initial and target orbits	153
7.5	Low-thrust Lunar transfer dynamics constants and Q-Law parameters.	154
7.6	Mars transfer and capture dynamics constants and problem parameters.	157
7.7	Mars transfer and capture results. TOFs refer to the spiral phase only.	157
7.8	Lunar swingby escape dynamics constants and problem parameters.	161
7.9	Lunar gravity-assist and comet flyby trajectory results. TOFs refer to the spiral phase only.	161

List of Figures

2.1	Classical orbital elements.	16
2.2	Logistic function drop off.	21
2.3	Eclipse diagram.	22
3.1	Typical Q-Law optimization setup.	31
3.2	LEO-GEO trajectories.	33
3.3	Equatorial to polar trajectory with unity gains and no coasting. Flight time = 33.3 days.	36
3.4	Equatorial to polar trajectory with weighted a, e and no coasting. Flight time = 42.3 days	37
3.5	Eccentric transfer trajectory with initial and final orbits included. . .	40
3.6	Baseline solution osculating orbital elements.	40
3.7	$W_\theta = 10$ solution osculating orbital elements.	41
3.8	Osculating Q values.	42
3.9	$\overline{M}_0 = 0$ targeting solution osculating orbital elements.	44
4.1	Q-Law and NLP problem setup.	47
4.2	Q-Law solution from GTO-GEO.	63
4.3	GTO-GEO solution for the 50% case. The Q-Law phase appears in red and the GPOPS-II phase in yellow.	64
4.4	GPOPS-II solution from GTO-GEO.	65
4.5	Time-optimal osculating orbital elements.	66
4.6	Mass-optimal and Q-Law effectivity equatorial $x-y$ trajectory projec- tions for the GTO-GEO 50% Case. Optimal/forced coasts appear in blue.	71
4.7	Normalized thrust magnitude for mass-optimal solution and Q-Law with effectivity checks. The first 60 days are omitted as they are the same for each case.	72
5.1	Forward and backward Q-Law comparison.	77
5.2	Forward-Backward Q-Law depicted in the Earth-Moon rotating frame. .	78
5.3	Time-optimal SMART-I trajectory in ECI coordinates.	85
5.4	Mass-optimal SMART-I trajectory in ECI coordinates.	86
5.5	Mass-optimal SMART-I trajectory in rotating selenocentric coordi- nates.	87
5.6	Forward-Backward Q-Law optimization procedure.	91

5.7	Results of the ESPA-Class Mission departure right ascension and epoch sweep.	94
5.8	Minimum time solutions for the ESPA-Class Mission with a January 1 departure.	95
5.9	Mass and epoch errors for the trajectories shown in Figure 5.8.	95
5.10	Minimum time solutions for a January 1 departure with right ascension = 90°	96
5.11	ESPA-Class Mission Final Mass vs. TOF.	98
5.12	Earth-centered mass-optimal trajectory in the inertial frame.	99
5.13	Selenocentric mass-optimal trajectory in the rotating frame.	100
5.14	All Backward Q-Law optimization procedure.	101
5.15	Backward Q-Law solutions for the LEO-LLO scenario.	104
6.1	A single Sims-Flanagan phase divided into N time steps.	108
6.2	Planetary gravity assist.	110
6.3	Monotonic Basin Hopping procedure.	111
6.4	Double Swingby trajectory design process.	114
6.5	Families of low-thrust Moon-Moon transfers to Comet 45P identified by MBH. Transfers are shown for $v_{\infty 0} = 0.5, 0.6$, and 0.7 km/s.	118
6.6	Sample escape trajectories to Comet 45P.	119
6.6	Sample escape trajectories to Comet 45P.	120
6.7	Comet 45P flyby mass trade for the spiral escape, single Lunar gravity assist, and double Lunar gravity assist cases.	121
6.8	Single gravity assist solutions in the ECI frame.	122
6.9	Earth-centered E- r_p Tisserand plot with before and after flyby states included. LGA ₁ and LGA ₂ refer to the first and second Lunar gravity assists, respectively.	123
7.1	Single shooting.	138
7.2	Multiple shooting.	139
7.3	Q-Law multiple shooting optimization procedure.	144
7.4	Logistic function coasting at $150^\circ \geq \theta \leq 210^\circ$ with $c_\theta = 100$	147
7.5	GTO-GEO Trajectory solutions. The NLP fixed-step integrator solution with low-fidelity eclipsing is shown on the top. The high-fidelity solution is shown on the bottom.	152
7.6	Backward Q-Law shooting problem setup to target velocity-vector spiral terminal state.	154
7.7	Low-thrust transfer to the Moon in the Earth-centered inertial frame (top) and the selenocentric rotating frame (bottom).	155
7.8	Sims-Flanagan + Q-Law Shooting problem setup for a Mars transfer and capture.	156
7.9	Low-thrust Martian capture trajectories.	158
7.9	Low-thrust Martian capture trajectories.	159

7.10	Sims-Flanagan + Q-Law Shooting problem setup for a Lunar swingby spiral escape. Comet Image: ESA/Rosetta/NAVCAM, CC BY-SA IGO 3.0	160
7.11	Low-thrust escape trajectories to Comet 45P.	162
7.11	Low-thrust escape trajectories to Comet 45P.	163

Chapter 1: Introduction

1.1 Problem Definition and Motivation

In recent years, the use of low-thrust solar electric propulsion (SEP) has become increasingly common in both Earth-orbiting and interplanetary spacecraft due to the specific impulse improvements of roughly one order of magnitude over chemical alternatives. This increased efficiency results in smaller launch vehicles, larger dry masses to orbit, longer propellant-limited lifetimes, and/or more capable trajectory designs. However, these transfers require higher electric power (typically generated by larger solar arrays) and longer transfer times. In particular, planetocentric low-thrust transfers between widely spaced orbits often require hundreds of orbital revolutions and last many months. Generating many-revolution low-thrust trajectories presents a challenging design problem since the directions of the thrust vector must be solved throughout each orbit. Furthermore, when SEP systems are chosen, eclipsing effects must also be considered. During an eclipse, the spacecraft's solar panels are shaded from the Sun and therefore cannot power the thruster, forcing a coast arc. This constraint introduces discontinuous dynamics and further increases the complexity of the already-challenging many-revolution optimal transfer problem.

Application of these low-thrust propulsion systems is becoming increasingly popular as space agencies look to leverage SmallSat spacecraft for Lunar and interplanetary exploration. The SmallSat mission class is particularly relevant to NASA’s SIMPLEx program and the interplanetary rideshare mission concept in general, where lower-budget missions enjoy a discounted launch cost and ride secondary to a much larger mission. Unfortunately, rideshare opportunities to the Moon or interplanetary space are much less common and provide little scheduling flexibility to the secondary payloads as compared to Geostationary Transfer Orbit (GTO) rideshares. If equipped with a low-thrust propulsion system, a rideshare spacecraft placed in GTO could spiral out to a capture orbit at the Moon or perform Lunar gravity assist(s), therefore raising its Earth escape velocity for a more efficient interplanetary cruise. The SMART-1 mission demonstrated this concept by leveraging a rideshare into GTO, from which it spiraled out over many months and then captured at the Moon [1, 2]. The Lunar Trailblazer SIMPLEx mission also leverages this framework to achieve a 100 km polar orbit at the Moon [3].

Low-thrust Lunar missions typically involve ESPA-class spacecraft, as ESPA ring adapters are commonly used and are often a proposed architecture for rideshare missions [4]. The SMART-1, Lunar Trailblazer, and Lunar Ice Cube spacecraft all classify as ESPA-class and employ low-thrust engines [2, 3, 5]. These missions demonstrate that low-cost Lunar exploration is possible using SmallSats but requires low-thrust engines to achieve Lunar science orbits. The GTO rideshare mission concept can be enabling for the SmallSat community, as it eliminates the dependence on larger interplanetary/Lunar launches while still yielding significant launch cost

reductions. The drawbacks of this mission type are much longer flight times and a challenging trajectory design problem to solve. Unfortunately, the problem complexity and computational burden can force analysts to generate only locally optimal point solutions rather than developing insight into the global trade space.

The objective of this work was to further develop an existing low-thrust guidance algorithm by improving solution optimality, algorithm efficiency, and the application/relevance to real-world mission design problems. Spiral Lunar transfers and gravity assist trajectories present complex examples of the many-revolution transfer problem, are highly desirable to space agencies and the SmallSat community. This work aimed to extend the application of this analytical guidance law to these trajectory types and provide an efficient method for exploring the trade space, giving analysts better insight to the mission design space. Additionally, significant effort was given to effectively pairing the guidance algorithm with other direct optimization techniques to reduce computation time and generate optimal high-fidelity solutions.

1.2 Low-Thrust Many-Revolution Design Methods

Historically, the many-revolution problem has been solved with indirect optimal control techniques [6, 7]. Indirect optimization uses the necessary conditions from the calculus of variations to formulate a two-point boundary value problem. Typically numerical techniques are used to aid in the solution of this problem. Indirect methods are advantageous in that the problem dimension remains small, making

them suitable for a differential corrector scheme like single or multiple shooting. In this approach, a guess is provided for costate parameters, and the equations of motion are evaluated. The costates are then corrected by the numerical algorithm employed, and the process is repeated until the optimality conditions are satisfied. Unfortunately, initial guesses for the costates are nonintuitive, and the resulting trajectory is very sensitive to these parameters. This sensitivity is further amplified for more complex trajectories with many revolutions. Indirect methods also suffer from the need to re-derive the equations of motion for the costates for differing dynamical environments, which can be challenging when trying to implement a high-fidelity dynamical model.

One common approach to reduce overall sensitivity is orbital averaging, where a new set of approximate dynamics are obtained by eliminating the fast variable and representing incremental changes in the orbital elements over each revolution [8, 9]. Orbital averaging is computationally efficient and an excellent approach for initial planning due to accurate time-of-flight (TOF) and mass prediction. Adversely, omission of the fast variable in the optimization can lead to trajectory inaccuracy in dynamical regimes in which averaged behavior is not representative. This can occur for trajectories with a very large semi-major axis, as the uncertainty on the trajectory solution can have more significant consequences. For example, when trying to rendezvous with the Moon, the period of the Lunar intercept orbit is a significant fraction of the Lunar synodic period. Uncertainty on this orbit due to averaged approximations will have major repercussions when trying to properly phase with the Moon for intercept.

Recent contributions by Aziz introduced the Sundman transformation to Differential Dynamic Programming (DDP) to solve for many-revolution trajectories [10]. DDP is based on Bellman’s Principle of Optimality [11], and it involves dividing the trajectory into multiple stages and solving an optimization subproblem for each stage. The subproblems are solved in a backward sweep from the final to the initial stage, with each subproblem solution minimizing the *cost-to-go* incurred from the subsequent trajectory stages. After the backward sweep identifies the controls for each stage, a forward sweep integrates the states and controls forward to obtain a new reference trajectory for the next backward sweep. This process is repeated until convergence.

Direct methods are another common technique for producing spiral trajectories, which involve discretizing the continuous optimal control problem into a Nonlinear Programming Problem (NLP). While DDP solves a small NLP for each stage subproblem, direct methods solve one large NLP for the entire trajectory. An example of this approach is Runge-Kutta (RK) parallel shooting with equinoctial elements, which was used to generate time-optimal spiral trajectories by Scheel and Conway [12]. Another popular technique is collocation. In this approach, the discretization is accomplished by representing the states and control parameters with polynomial splines and using either integral or derivative constraints to enforce the system dynamics [13, 14, 15]. This approach was used extensively within this work.

To handle the discontinuous SEP control dynamics caused by eclipses, Geffroy and Epenoy used an averaging approach to incorporate environment constraints like eclipsing from Edelbaum’s cylindrical shadow model [16, 17]. Ferrier and Epenoy

were among the first to exploit smoothing techniques for eclipses while studying the indirect problem [18]. Betts and Graham and Rao used a direct collocation method to solve for many-revolution trajectories with eclipsing [19, 20]. In both of their approaches, trajectory thrust phases were terminated at the shadow entrance, and a coast phase was enforced until the shadow exit. Aziz developed a smooth logistic function to model eclipses with DDP [21]. The logistic function method was adapted to direct collocation and multiple shooting in this work.

Unfortunately, these techniques, and trajectory optimization techniques as a whole for the many-revolution problem, are either computationally sensitive, expensive, or rely on averaged dynamics. However, heuristic guidance algorithms have been developed as a means to rapidly generate high-fidelity, close-to-optimal trajectories, most notably, Petropoulos’s Q-Law [22, 23]. Q-law is a feedback, Lyapunov control method with an analytical thrust vector calculation. Q-Law, although typically sub-optimal, produces trajectories very quickly and includes the fast variable in the system dynamics. The user can influence Q-Law’s behavior and performance by tuning sets of gains that relate the importance of different orbit element targets. Classical Q-Law’s primary flaw arises when trying to target a full 6-state boundary condition because Q-Law asymptotically approaches the desired orbit and cannot target a specific true anomaly value. In the past, Q-Law solutions have been used as an initial guess in the high-fidelity DDP trajectory optimization software Mystic [24]. The aspects of all of the preceding approaches are briefly summarized in Table 1.1.

Table 1.1: Qualitative aspects of many-revolution low-thrust transfer methodologies.

Method	Optimality	Sensitivity	Problem Size	Dynamics	Full State Targeting
Indirect[6, 7, 18]	Local	High	Low	Full	Yes
Direct[20]	Local	Moderate/High	High	Full	Yes
Averaged[8, 9]	Local	Low	Low	Approximate	No
DDP [10]	Local	Moderate/High	High	Full	Yes
Closed-loop[22, 23]	Sub-optimal	Very Low	Low	Full	No

1.3 Dissertation Overview

The focus of this work is to extend the Q-Law guidance algorithm relevance and applicability to realistic mission design scenarios. To accomplish this, the classical Q-Law method is combined with other mission design software and direct optimization techniques and used to generate complex planetocentric, Cislunar, and interplanetary trajectories. This research is presented in the following manner:

- **Chapter 1:** The introductory chapter begins with an overview of low-thrust transfers and their application to modern space exploration. A literature review of historical approaches is provided. A comparison is given for these methods, and the benefits of analytical guidance laws are identified.
- **Chapter 2:** The dynamical model used throughout this research is presented. This chapter begins with the governing equations for perturbed orbital motion, which includes the perturbation models for spacecraft thrust, third body gravity, and an aspherical central body. Next, the classical and modified equinoctial orbital element sets are presented. A smooth eclipse model for solar electric propulsion spacecraft is given.
- **Chapter 3:** The guidance algorithm used in this research, Q-Law, is pre-

sented. This chapter begins with a brief overview of Lyapunov control theory and presents the Q-Law thrust vector calculation. Practical strategies for implementing Q-Law, like gain tuning, are discussed. Simple example problems that demonstrate the benefits of using coast arcs and gain tuning are given. Sixth element targeting challenges are also discussed.

- **Chapter 4:** This chapter focuses on a hybrid method that combines Q-Law with direct collocation. First, the theory behind direct collocation and nonlinear programming are briefly given. Also, the numerical Jacobian calculation techniques relevant to this research are discussed. An example low-thrust transfer from GTO to Geostationary orbit (GEO) is solved using the methods presented. Time-optimal and mass-optimal trajectories are produced, and the results are compared to existing literature solutions.
- **Chapter 5:** A new approach that combines forward and backward propagated Q-Law is presented and applied to spiral Lunar transfers. This approach is demonstrated as an initial guess tool for a test problem inspired by the SMART-1 mission. Numerical results are compared to existing literature. This method is then demonstrated as a trade study tool for a realistic Earth-Moon transfer. A departure orbit parameter sweep is performed and the resulting Q-Law solutions are used as an initial guess for direct optimization. Lastly, fixed arrival condition transfers from Low-Earth Orbit (LEO) to Low-Lunar Orbit (LLO) are solved using only backward propagated Q-Law. Propellant usage trade studies are generated for this scenario.

- **Chapter 6:** Spiral Lunar swingby escape trajectories are explored. Q-Law is paired with the Evolutionary Mission Trajectory Generator (EMTG) to produce single and double Lunar gravity assist trajectories for a SmallSat spacecraft traveling to a comet.
- **Chapter 7:** Analytical partial derivatives of the Q-Law thrust vector calculation are derived. A smooth true anomaly coasting function is also given. The process of using these derivatives to produce the trajectory state transition matrix is presented and a new Q-Law shooting technique is formulated. This method is demonstrated on several test problems, which include a GTO-GEO transfer, an Earth-Moon transfer, a Mars transfer and spiral down, and a single Lunar gravity assist escape.
- **Chapter 8:** A summary of this work is presented. Recommendations for future work are proposed.

1.4 Summary of Contributions

The major contributions of this thesis to the state of the art are:

- A demonstration of the benefits of combining Q-Law with direct collocation for many-revolution trajectories. A logistic eclipse model was incorporated into the collocation for smooth-eclipse handling.
- The development of a Forward-Backward Q-Law approach for rapid low-thrust Lunar trajectory design. This method is shown to produce effective initial

guesses for optimization and can be used for preliminary trade study analysis.

- The exploration of SmallSat spiral escape trajectories using backward Q-Law and the perturbed Sims-Flanagan model.
- The derivation and application of analytical partial derivatives of Q-Law thrust vector calculation to compute the trajectory state transition matrix. A Q-Law shooting nonlinear programming problem is formulated for gradient-based gain tuning and to enforce nonlinear constraints on the initial state.
- The combination of Q-Law shooting with the Sims-Flanagan interplanetary model for end-to-end trajectory design in one optimization problem.

Chapter 2: System Dynamics

2.1 Perturbed Orbital Motion

The most basic spacecraft trajectory occurs in the two-body problem, when one body has negligible mass and the other is a point mass central body. In this scenario, the spacecraft motion can be described by Kepler's laws of planetary motion as listed in Reference [25] and described below:

1. The orbit of each planet is an ellipse with the Sun at one focus.
2. The line joining the planet to the Sun sweeps out equal areas in equal times.
3. The square of the period of a planet is proportional to the cube of its mean distance to the Sun.

The spacecraft state can be written in cartesian coordinates as shown below, with boldface variables representing a vector quantity.

$$\mathbf{r} = [x, \ y, \ z]^T \tag{2.1}$$

$$\mathbf{v} = [\dot{x}, \ \dot{y}, \ \dot{z}]^T \tag{2.2}$$

In this state representation, the equations of motion for a Keplerian orbit are

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \quad (2.3)$$

with μ being the central body gravitational constant (GM) and $r = \|\mathbf{r}\|$.

In many cases, the two-body problem is an over-simplification of the system dynamics, and perturbations are included to improve the orbit fidelity. Perturbations represent the other forces acting on the spacecraft besides the point mass central body gravity. These forces can include aspherical gravity of the central body, gravitational forces from other planetary bodies, solar radiation pressure, and thrust from the spacecraft's propulsion system. When perturbations are included, the cartesian two-body dynamics become

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \boldsymbol{\delta} \quad (2.4)$$

where $\boldsymbol{\delta}$ represents the sum of the inertial perturbing accelerations. In this work, perturbations from J_{2-4} and n-body gravity were included in addition to the thrust generated by the spacecraft's low-thrust propulsion system.

The aspherical gravity perturbations are calculated using Equations 2.5 to 2.8.

$$\delta g_r = -\frac{\mu}{r^2} \sum_{k=2}^n (k+1) \left(\frac{R_e}{r}\right)^k P_k(\sin \phi) J_k \quad (2.5)$$

$$\delta g_n = -\frac{\mu \cos \phi}{r^2} \sum_{k=2}^n (k+1) \left(\frac{R_e}{r}\right)^k P'_k(\sin \phi) J_k \quad (2.6)$$

$$\hat{\mathbf{i}}_n = \frac{\hat{\mathbf{e}}_n - (\hat{\mathbf{e}}_n^T \hat{\mathbf{i}}_r) \hat{\mathbf{i}}_r}{\left\| \hat{\mathbf{e}}_n - (\hat{\mathbf{e}}_n^T \hat{\mathbf{i}}_r) \hat{\mathbf{i}}_r \right\|} \quad (2.7)$$

$$\delta g = \delta g_n \hat{\mathbf{i}}_n - \delta g_r \hat{\mathbf{i}}_r \quad (2.8)$$

Here, P'_k is the derivative of the k^{th} Legendre polynomial, P_k , with respect to $\sin \phi$, $\hat{\mathbf{e}}_n = [0, 0, 1]^T$, ϕ is the geocentric latitude, R_e is the Earth's radius, J_k are the zonal harmonic coefficients, and $\hat{\mathbf{i}}_r$ is the spacecraft position unit vector, defined later in Equation 2.14. Additionally, third-body accelerations are calculated in the inertial frame using Equation 2.9, where \mathbf{r}_{j0} is the position of the j^{th} body with respect to Earth and \mathbf{r}_j is the position of the spacecraft with respect to the j^{th} body.

$$\delta p = \sum_{j=1}^{N_b} -\mu_j \left(\frac{\mathbf{r}_{j0}}{r_{j0}^3} + \frac{\mathbf{r}_j}{r_j^3} \right) \quad (2.9)$$

The spacecraft thruster can be throttled and is modeled using a maximum flow rate \dot{m}_{\max}

$$\dot{m}_{\max} = -\frac{T_{\max}}{g_0 I_{\text{sp}}} \quad (2.10)$$

where T_{\max} is the maximum thrust, I_{sp} is the specific impulse, and g_0 is standard gravity. The actual flow rate is then

$$\dot{m} = \dot{m}_{\max} \sqrt{\mathbf{T}_n \cdot \mathbf{T}_n + \lambda^2} \quad (2.11)$$

where \mathbf{T}_n is the normalized thrust vector.

In Equation 2.11, λ is used as a fictitious “mass leak” that prevents the undefined mass flow rate partial derivatives that arise when the magnitude of normalized thrust approaches zero. Typically, λ is set to be a small number that does not significantly impact solution accuracy but permits optimal coast arcs in gradient-based optimization. If the spacecraft thrust vector is represented in the inertial frame, then the perturbing acceleration vector due to thrust and the total inertial disturbing acceleration is calculated using Equations 2.12 and 2.13.

$$\delta \mathbf{T} = \frac{T}{m} \mathbf{T}_n \quad (2.12)$$

$$\delta = \delta g + \delta p + \delta T \quad (2.13)$$

where T is the thrust applied and m is the spacecraft mass.

2.2 Orbital Element Sets

It is often beneficial to use orbital element sets instead of the cartesian representation because most of the elements vary slowly when perturbations are present. In this work, classical orbital elements (COE) and modified equinoctial elements (MEE) were utilized. The equations of motion for both of these element sets are described by variational equations defined in the spacecraft’s local frame. This frame is formed by the basis vectors $(\hat{\mathbf{i}}_r, \hat{\mathbf{i}}_\theta, \hat{\mathbf{i}}_h)$.

$$\hat{\mathbf{i}}_r = \frac{\mathbf{r}}{\|\mathbf{r}\|}, \quad \hat{\mathbf{i}}_h = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|}, \quad \hat{\mathbf{i}}_\theta = \hat{\mathbf{i}}_h \times \hat{\mathbf{i}}_r \quad (2.14)$$

For use with either orbital element set, the perturbations described in Equations 2.5 to 2.9 must be rotated into the local frame using the transformation formed by the frame's basis vectors, as shown in Equations 2.15 to 2.17:

$${}^R\mathbf{Q}^I = \begin{bmatrix} \hat{\mathbf{i}}_r & \hat{\mathbf{i}}_\theta & \hat{\mathbf{i}}_h \end{bmatrix}^T \quad (2.15)$$

$$\Delta_g = {}^R\mathbf{Q}^I \delta \mathbf{g} \quad (2.16)$$

$$\Delta_p = {}^R\mathbf{Q}^I \delta \mathbf{p} \quad (2.17)$$

The total disturbing acceleration vector in the local frame $\Delta = (\Delta_r, \Delta_\theta, \Delta_h)$ is then modeled as

$$\Delta = \Delta_T + \Delta_g + \Delta_p \quad (2.18)$$

where Δ_T is similar to Equation 2.12 but with the thrust vector represented in the local frame, Δ_g is the central body's higher order gravity acceleration, and Δ_p represents third-body accelerations. Typically, the control law or optimizer determines \mathbf{T}_n directly in the local frame, so no rotation is needed to calculate Δ_T .

2.2.1 Classical Orbital Elements

The classical orbital elements provide an easy-to-visualize representation of the osculating orbit. This set is comprised of semi-major axis a , eccentricity e , inclination i , argument of periapsis ω , and right ascension of the ascending node Ω . Semi-major axis and eccentricity describe the size and shape of the orbit, and the other elements dictate the orientation of the orbit plane. One additional quantity, such as true anomaly θ , specifies the spacecraft's location on the orbit. Some of these elements are illustrated in Figure 2.1. The drawback of the classical element set is the singularities at circular and equatorial orbits. When $e = 0$, ω and θ are undefined, and when $i = 0$, Ω and ω are undefined.

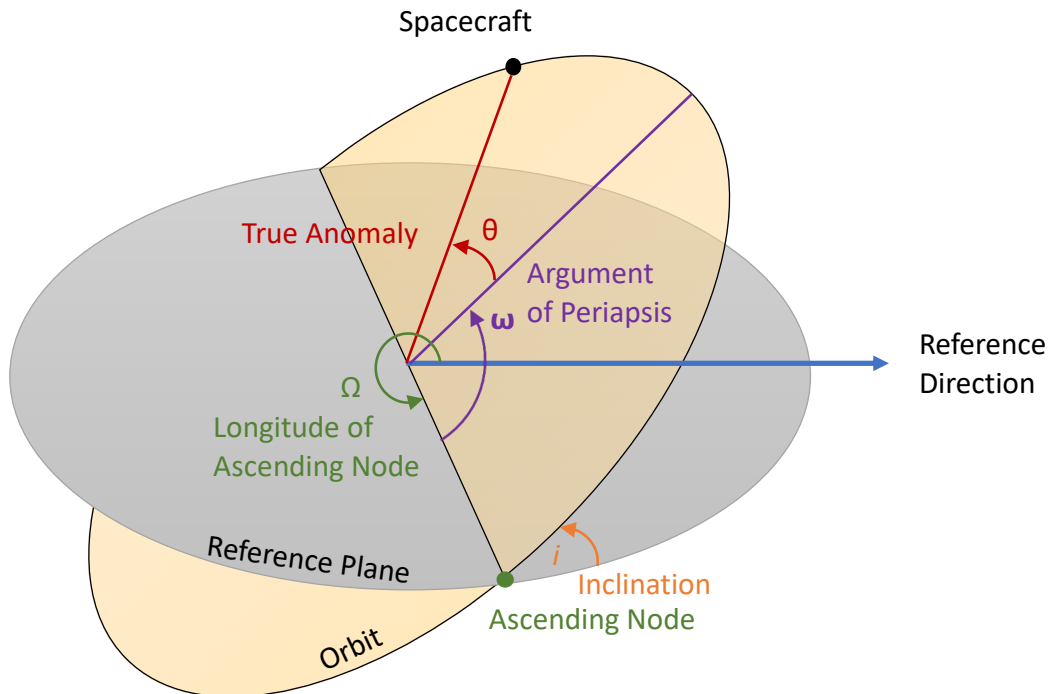


Figure 2.1: Classical orbital elements.

Perturbations can be applied to the classical orbital elements through the variational equations shown in Equation 2.19. These equations are defined in the spacecraft's local frame formed by the basis vectors $(\hat{\mathbf{i}}_r, \hat{\mathbf{i}}_\theta, \hat{\mathbf{i}}_h)$.

$$\begin{aligned}
\dot{a} &= \frac{2a^2}{h} \left(e \sin \theta \Delta_r + \frac{p}{r} \Delta_\theta \right) \\
\dot{e} &= \frac{1}{h} \left(p \sin \theta \Delta_r + [(p+r) \cos \theta + re] \Delta_\theta \right) \\
\dot{i} &= \frac{r \cos(\theta + \omega)}{h} \Delta_h \\
\dot{\Omega} &= \frac{r \sin(\theta + \omega)}{h \sin i} \Delta_h \\
\dot{\omega} &= \frac{1}{eh} \left(-p \cos \theta \Delta_r + (p+r) \sin \theta \Delta_\theta \right) - \frac{r \sin(\theta + \omega)}{h \tan i} \Delta_h \\
\dot{\theta} &= \frac{h}{r^2} + \frac{1}{eh} \left(p \cos \theta \Delta_r - (p+r) \sin \theta \Delta_\theta \right)
\end{aligned} \tag{2.19}$$

where h is the specific angular momentum, p is the semi-latus rectum, r is the current orbital radius, and Δ_r , Δ_θ , Δ_h are the perturbing acceleration components in the local frame. These perturbing accelerations can be decomposed to in-plane and out-of-plane angles α and β and the specific acceleration magnitude f .

$$\begin{aligned}
\Delta_r &= f \cos \beta \sin \alpha, \\
\Delta_\theta &= f \cos \beta \cos \alpha, \\
\Delta_h &= f \sin \beta.
\end{aligned} \tag{2.20}$$

The full COE state also includes the spacecraft mass:

$$\mathbf{x}_{COE} = [a, \ e, \ i, \ \omega, \ \Omega, \ \theta, \ m]^T \quad (2.21)$$

2.2.2 Modified Equinoctial Elements

In this work, some trajectory optimization efforts employed a modified equinoctial element state representation with spacecraft mass included [26, 27]. The MEE set, \mathbf{l} , includes the semi-latus rectum, p , four parameters describing eccentricity and inclination, f, g, h, k , and the true longitude fast variable, L .

$$\mathbf{l} = [p, \ f, \ g, \ h, \ k, \ L]^T \quad (2.22)$$

The relationship between MEE and the classical orbital elements is

$$\begin{aligned} p &= a(1 - e^2) \\ f &= e \cos(\omega + \Omega) \\ g &= e \sin(\omega + \Omega) \\ h &= \tan(i/2) \sin \Omega \\ k &= \tan(i/2) \cos \Omega \\ L &= \theta + \omega + \Omega \end{aligned} \quad (2.23)$$

The corresponding equations of motion for an MEE state perturbed by an acceler-

ation are expressed below:

$$\dot{\mathbf{i}} = A\Delta + \mathbf{b} \quad (2.24)$$

Here, A is a 6 x 3 matrix defined as:

$$A = \frac{1}{q} \sqrt{\frac{p}{\mu}} \begin{bmatrix} 0 & 2p & 0 \\ q \sin L & (q+1) \cos L + f & -g(h \sin L - k \cos L) \\ -q \cos L & (q+1) \sin L + g & f(h \sin L - k \cos L) \\ 0 & 0 & \frac{1}{2}(1 + h^2 + k^2) \cos L \\ 0 & 0 & \frac{1}{2}(1 + h^2 + k^2) \sin L \\ 0 & 0 & h \sin L - k \cos L \end{bmatrix} \quad (2.25)$$

where q is a constant used to simplify the expression:

$$q = 1 + f \cos(L) + g \sin(L) \quad (2.26)$$

The vector \mathbf{b} is the effect of the central-body's point-mass acceleration on the MEE state representation's fast variable:

$$\mathbf{b} = \left[0, \ 0, \ 0, \ 0, \ 0, \ \sqrt{\mu p} \left(\frac{q}{p} \right)^2 \right]^T \quad (2.27)$$

The full MEE state also includes the spacecraft mass:

$$\mathbf{x}_{MEE} = [p, f, g, h, k, L, m]^T \quad (2.28)$$

The trajectory optimization also employed the Sundman transformation [28] to transform the independent variable from time to true longitude, L . This transformation reduces the sensitivity of the many-revolution problem, and when beginning with a mesh equally spaced in true longitude, can reduce the number of mesh refinements needed. Differentiation with respect to true longitude is denoted by ($'$).

$$t' = \frac{\partial t}{\partial L} = \dot{L}^{-1} \quad (2.29)$$

The state dynamics then become

$$\mathbf{x}' = \dot{L}^{-1} [\dot{p}, \dot{f}, \dot{g}, \dot{h}, \dot{k}, 1, \dot{m}]^T \quad (2.30)$$

with the 1 included to bring time into the state vector in place of the fast variable.

2.3 Eclipse Model

The eclipsing model used within the trajectory optimization problems leveraged a logistic function and follows the method presented by Aziz *et al.* [21], as shown in Equations 2.31 to 2.34. Here, $\mathbf{r}_{\odot/sc}$ is the position of the Sun with respect to the spacecraft, $\mathbf{r}_{B/sc}$ is the position of the central body with respect to the spacecraft, R_B is the central body radius, and R_{\odot} is the Sun's radius.

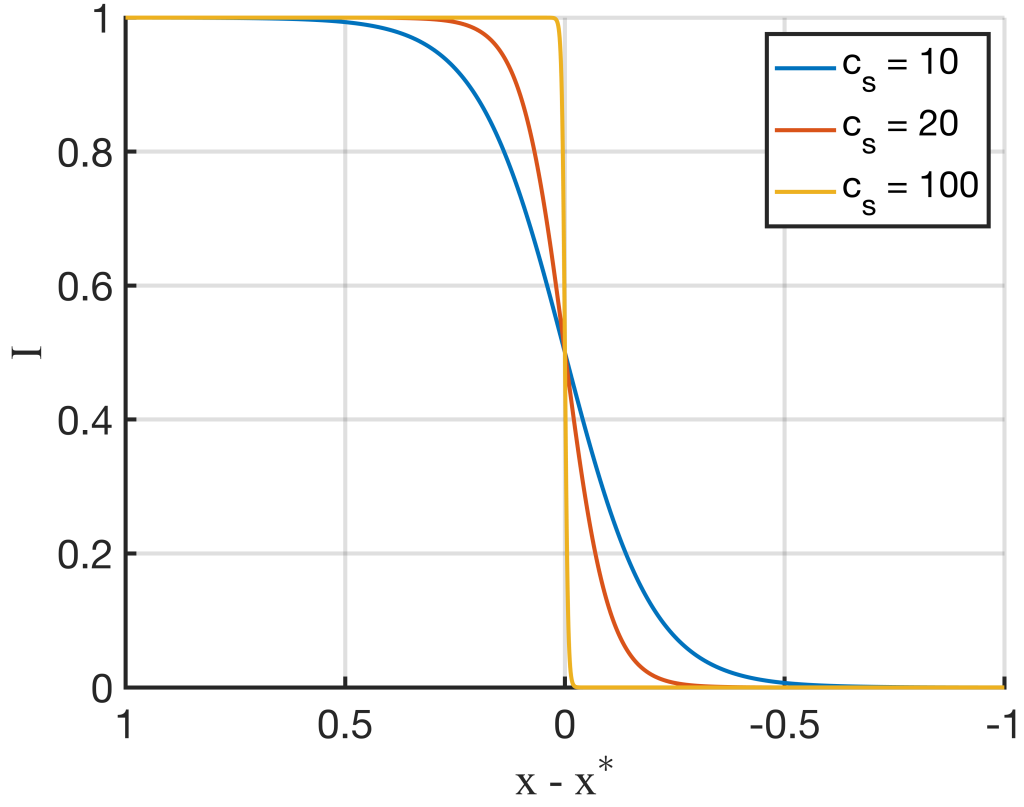


Figure 2.2: Logistic function drop off.

$$aSR = \arcsin \frac{R_{\odot}}{r_{\odot/sc}} \quad (2.31)$$

$$aBR = \arcsin \frac{R_B}{r_{B/sc}} \quad (2.32)$$

$$aD = \arccos \frac{\mathbf{r}_{B/sc}^T \mathbf{r}_{\odot/sc}}{r_{B/sc} r_{\odot/sc}} \quad (2.33)$$

$$I = \frac{1}{1 + e^{-c_s[aD - c_t(aSR + aBR)]}} \quad (2.34)$$

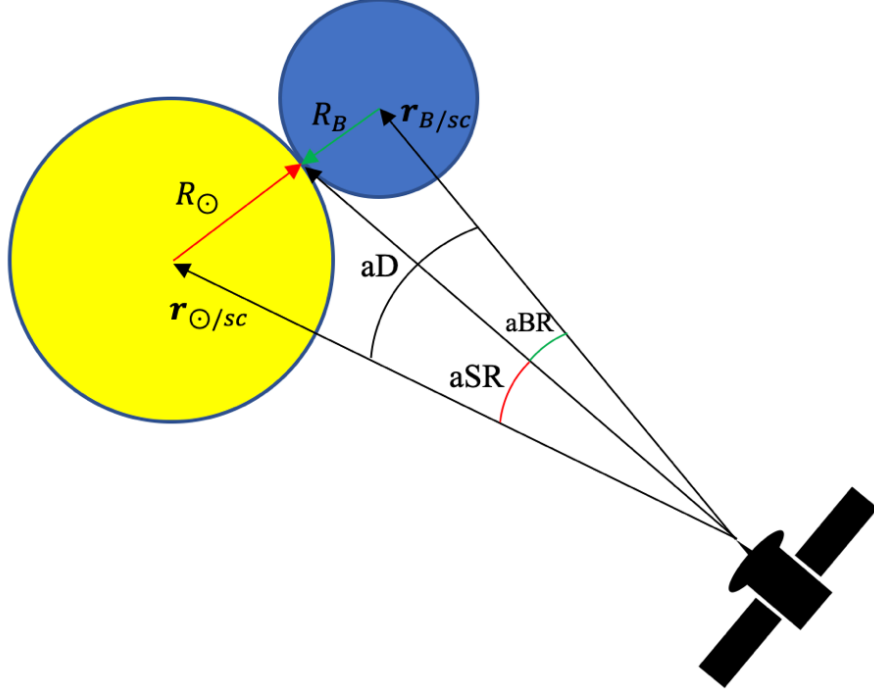


Figure 2.3: Eclipse diagram.

The spacecraft is eclipsed when $aSR + aBR > aD$. Figure 2.2 shows the reduction in function value at the event crossing, and Figure 2.3 shows the system at the beginning of an eclipse. Figure 2.3 was recreated in Reference [29] based on Reference [21]. The logistic function in Equation 2.34 provides penumbra detection and is used to limit the available power during both partial and full eclipses. The magnitude of I is bounded between 0 and 1. This function provides continuity when entering an eclipse, which is crucial for gradient-based optimization schemes that require continuous dynamics throughout each phase. The coefficients, c_s and c_t , are used to affect the sharpness of the intensity curve when passing in/out of the shadow. The nominal coefficient values presented by Aziz *et al.* were used, $c_s = 298.78$ and $c_t = 1$. When Q-Law propagation was not performed within a gradient-based optimization scheme, root-finding was used to exactly detect shadow crossings. The

power discontinuities were acceptable in this case as event functions stopped the integration to adjust the power available for thrusting. To determine the ephemeris position of the Sun and other planetary bodies, the SPICE Toolkit and the de430 planetary ephemeris kernel were used [30]. The trajectory optimization eclipsing constraint is formulated as

$$\Delta_T = \frac{T_{\max}}{m} I \mathbf{T}_n \quad (2.35)$$

Chapter 3: Q-Law Lyapunov Guidance Algorithm

Heuristic guidance algorithms have been developed as a means to rapidly generate close-to-optimal trajectories, with the most notable being Petropoulos's Q-Law [22, 23]. Q-law is a Lyapunov control method that determines the instantaneous thrust direction at each point along the trajectory toward the target orbit. The following section provides an introduction to Lyapunov control theory and presents the Q-Law guidance algorithm. The benefits of including coast arcs and gain tuning as well as sixth element targeting challenges are discussed and demonstrated.

3.1 Lyapunov Control Functions

A Lyapunov function is a continuous scalar function $V(\mathbf{x})$ with continuous first-order partial derivatives on a region D . It also must be positive definite everywhere except the origin, $V(\mathbf{x}) > 0$ for all $x \neq 0$. The stability of the system described by Equation 3.1 can be determined by the time-derivative of $V(\mathbf{x})$ [31].

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{3.1}$$

$$\dot{V}(\mathbf{x}) = \nabla V(\mathbf{x})^T \cdot \mathbf{f}(\mathbf{x}) \quad (3.2)$$

If there exists a Lyapunov function such that $\dot{V}(\mathbf{x}) \leq 0$ on some region D that contains the origin, then the system is stable. Furthermore, if $\dot{V}(\mathbf{x}) < 0$ on some region D that contains the origin, then the existence of $V(\mathbf{x})$ guarantees asymptotic stability of the origin. Q-Law's functionality comes from Lyapunov control theory, which extends the Lyapunov function concept to systems with controls inputs, as in Equation 3.3.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad \mathbf{u}(t) \in \mathbb{U} \quad (3.3)$$

A control-Lyapunov function for this system is a continuously differentiable scalar function on a region D that is positive definite except at the origin with the following property:

$$\forall \mathbf{x} \neq 0, \exists \mathbf{u} \quad \dot{V}(\mathbf{x}, \mathbf{u}) = \nabla V(\mathbf{x})^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{u}) < 0 \quad (3.4)$$

The control profile $\mathbf{u}(t)$ is then selected such that it minimizes $\dot{V}(\mathbf{x}, \mathbf{u})$ at each point in time, therefore reducing $V(\mathbf{x})$ toward the origin as quickly as possible.

$$\mathbf{u}^*(t) = \underset{\mathbf{u} \in \mathbb{U}}{\operatorname{argmin}} \dot{V}(\mathbf{x}, \mathbf{u}) \quad (3.5)$$

3.2 Q-Law Thrust Vector Calculation

The Q-Law candidate Lyapunov function is defined as

$$Q = (1 + W_P P) \sum_{\mathfrak{e}} W_{\mathfrak{e}} S_{\mathfrak{e}} \left[\frac{d(\mathfrak{e}, \mathfrak{e}_T)}{\dot{\mathfrak{e}}_{xx}} \right]^2 \quad (3.6)$$

with $\mathfrak{e} = (a, e, i, \omega, \Omega)$. In Equation 3.6, $\dot{\mathfrak{e}}_{xx}$ is the maximum rate of change of each element over the local frame thrust vector and true anomaly, θ . Also, $W_{\mathfrak{e}}$ represents the components of the gain vector \mathbf{W} given to the orbital elements, and $d(\mathfrak{e}, \mathfrak{e}_T)$ is the difference between the values of the current and target orbital elements. The maximum rate of change of each element, $\dot{\mathfrak{e}}_{xx}$, are

$$\dot{a}_{xx} = 2f \sqrt{\frac{a^3(1+e)}{\mu(1-e)}} \quad (3.7)$$

$$\dot{e}_{xx} = \frac{2pf}{h} \quad (3.8)$$

$$\dot{i}_{xx} = \frac{pf}{h(\sqrt{1-e^2 \sin^2 \omega} - e|\cos \omega|)} \quad (3.9)$$

$$\dot{\Omega}_{xx} = \frac{pf}{h \sin i (\sqrt{1-e^2 \cos^2 \omega} - e|\sin \omega|)} \quad (3.10)$$

$$\dot{\omega}_{xx} = \frac{\dot{\omega}_{xxi} + b\dot{\omega}_{xxo}}{1+b} \quad (3.11)$$

where b is nominally 0.01, f is the specific acceleration due to thrust, and

$$\dot{\omega}_{xxi} = \frac{f}{eh} \sqrt{p^2 \cos^2 \theta_{xx} + (p + r_{xx})^2 \sin^2 \theta_{xx}} \quad (3.12)$$

$$\dot{\omega}_{xxo} = \dot{\Omega}_{xx} |\cos i| \quad (3.13)$$

$$\begin{aligned} \cos \theta_{xx} = & \left[\frac{1 - e^2}{2e^3} + \sqrt{\frac{1}{4} \left(\frac{1 - e^2}{e^3} \right)^2 + \frac{1}{27}} \right]^{1/3} \\ & - \left[-\frac{1 - e^2}{2e^3} + \sqrt{\frac{1}{4} \left(\frac{1 - e^2}{e^3} \right)^2 + \frac{1}{27}} \right]^{1/3} - \frac{1}{e} \end{aligned} \quad (3.14)$$

$$r_{xx} = \frac{p}{1 + e \cos \theta_{xx}} \quad (3.15)$$

In Equation 3.6, S_{α} are the components of \mathbf{S} . In the thrust vector calculation, \mathbf{S} and P represent penalty functions that keep semi-major axis and periapsis from growing too large or too small. They are defined as

$$\mathbf{S} = \begin{cases} \left[1 + \left(\frac{a - a_T}{ma_T} \right)^n \right]^{1/r} & \text{for } \alpha = a \\ 1 & \text{for } \alpha = e, i, \omega, \Omega \end{cases} \quad (3.16)$$

$$P = \exp \left[k \left(1 - \frac{r_p}{r_{pmin}} \right) \right] \quad (3.17)$$

with the nominal values of $m = 3$, $n = 4$, $r = 2$, and $k = 100$. The proximity quotient Q is differentiated to minimize \dot{Q} , therefore driving Q to zero as quickly as possible to reach the desired orbit. The derivative of Q can be calculated using the chain rule, as shown in Equation 3.18,

$$\dot{Q} = \sum_{\mathfrak{e}} \frac{\partial Q}{\partial \mathfrak{e}} \dot{\mathfrak{e}} \quad (3.18)$$

with \mathfrak{e} coming from Equation 2.19. \dot{Q} can be rewritten in the following form [32]:

$$\dot{Q} = D_1 \cos \beta \cos \alpha + D_2 \cos \beta \sin \alpha + D_3 \sin \beta \quad (3.19)$$

The terms D_1, D_2 and D_3 can be calculated by evaluating

$$D_1 = \sum_{\mathfrak{e}} \frac{\partial Q}{\partial \mathfrak{e}} \frac{\partial \dot{\mathfrak{e}}}{\partial \Delta_\theta} \quad (3.20)$$

$$D_2 = \sum_{\mathfrak{e}} \frac{\partial Q}{\partial \mathfrak{e}} \frac{\partial \dot{\mathfrak{e}}}{\partial \Delta_r} \quad (3.21)$$

$$D_3 = \sum_{\mathfrak{e}} \frac{\partial Q}{\partial \mathfrak{e}} \frac{\partial \dot{\mathfrak{e}}}{\partial \Delta_h} \quad (3.22)$$

To calculate the partial derivatives of Q , it is convenient to define the intermediate term \mathbf{V} with components defined as

$$V_{\mathfrak{e}} = S_{\mathfrak{e}} \left[\frac{d(\mathfrak{e}, \mathfrak{e}_T)}{\dot{\mathfrak{e}}_{xx}} \right]^2 \quad (3.23)$$

The partial derivatives of Q are then found to be

$$\frac{\partial Q}{\partial \mathfrak{e}} = W_P \frac{\partial P}{\partial \mathfrak{e}} \mathbf{W}^T \mathbf{V} + (1 + W_P P) \mathbf{W}^T \frac{\partial \mathbf{V}}{\partial \mathfrak{e}} \quad (3.24)$$

The optimal local frame thrust vector for reducing Q is then found by aligning the

thrust vector in the anti-direction defined by the D_{1-3} coefficients.

$$\mathbf{u} = \frac{-(D_2, D_1, D_3)^T}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (3.25)$$

Q-Law also has an optional mechanism for coasting known as “effectivity”. Effectivity compares the effectiveness of reducing Q at the current point on the osculating orbit, \dot{Q}_n , with the most effective point on the current osculating orbit, \dot{Q}_{nn} . These expressions are given as [23]

$$\eta_a = \frac{\dot{Q}_n}{\dot{Q}_{nn}} \quad (3.26)$$

$$\eta_r = \frac{\dot{Q}_n - \dot{Q}_{nx}}{\dot{Q}_{nn} - \dot{Q}_{nx}} \quad (3.27)$$

where η_a and η_r are the absolute and relative effectivity, respectively, and

$$\dot{Q}_{nn} = \min_{\theta} \dot{Q}_n \quad (3.28)$$

$$\dot{Q}_{nx} = \max_{\theta} \dot{Q}_n \quad (3.29)$$

Analytic expressions for \dot{Q}_{nn} and \dot{Q}_{nx} are not available and are generally multimodal, so numerical approaches must be employed. To solve for these terms, a simple grid search over 50 equally spaced true anomaly points along $\theta = [0, 2\pi)$ results in a rapid solution with sufficient accuracy. Given a user-specified absolute or relative effectivity cutoff, thrust is applied when the current effectivity is above the cutoff value and $\Delta_T = \mathbf{0}$ otherwise. Although effectivity checks provide a simple and

straightforward coast mechanism, they add computational expense to the trajectory propagation.

3.3 Gain Tuning

The W_{oe} term in the Q-Law Lyapunov function refers to the user specified gains for each orbital element, which determine the priority given to each orbital element during the thrust vector calculation. A gain of zero means that element can vary unconstrained. A very large gain yields a thrust direction that converges the element to the desired value as quickly as possible. This provides a useful way to shape the trajectory by controlling which elements are changed most quickly, and the chosen gain values can have significant effects on the resulting trajectory. Unfortunately, the gain space typically has many local optima. To find globally optimal solutions, evolutionary algorithms are typically employed to optimize the targeted gains and minimize the TOF cost function. Additionally, multi-objective genetic algorithms can be used to generate TOF-propellant usage pareto-fronts and prove to be a reliable means of optimizing both the Q-Law gains and effectivity parameters [33, 34]. A general Q-Law optimization setup is shown in Figure 3.1. One benefit of using evolutionary algorithms is the ease with which the population can be parallelized across multiple processors, which decreases computation time and/or increases the population size that can be evaluated in a given amount of time.

A new technique for gain tuning is to use gradient-based optimization to iden-

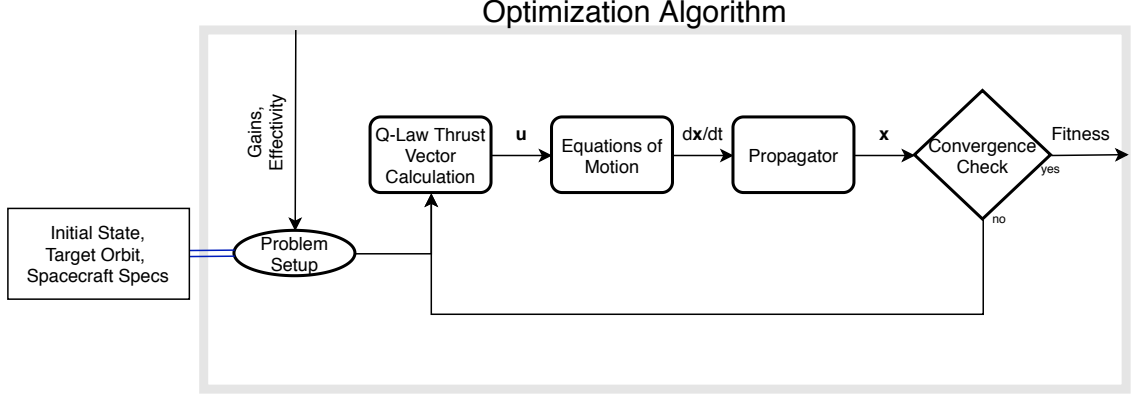


Figure 3.1: Typical Q-Law optimization setup.

tify locally optimal gain combinations. This method can be very effective at isolating the nearest local optima, and if a strong initial guess were provided, can be sufficient for finding suitable gains. This approach is introduced and developed in Chapter 7.

It is noted that there exists an MEE Q-Law formulation [32, 35, 36], but it is not used in this work due to the element coupling. In practice, it can be advantageous to let one or more of the classical elements vary unconstrained, such as the case when trying to target a GTO orbit without a fixed right ascension. However, due to the coupled relationships described in Equation 2.23, this cannot be accomplished using MEE Q-Law.

3.4 Basic Transfers

Q-Law was originally developed and tested on relatively simple low-thrust transfer problems. To demonstrate the function of the Q-Law gains and effectivity parameters, two simple test problems are solved. The first is a LEO to GEO transfer known as Case A in the original Q-Law manuscripts [22, 23], and the second is an

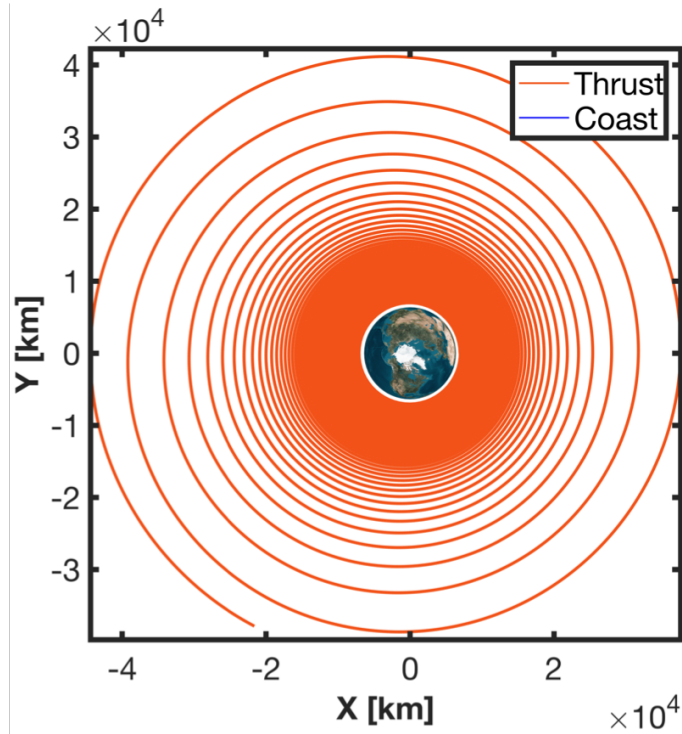
equatorial to polar orbit transfer that was used to test heuristic guidance laws by Hatten [37]. In both problems, the dynamics are reduced to the two-body problem without eclipsing. The problem parameters are listed in Tables 3.1 and 3.2.

Table 3.1: LEO-GEO problem specifications.

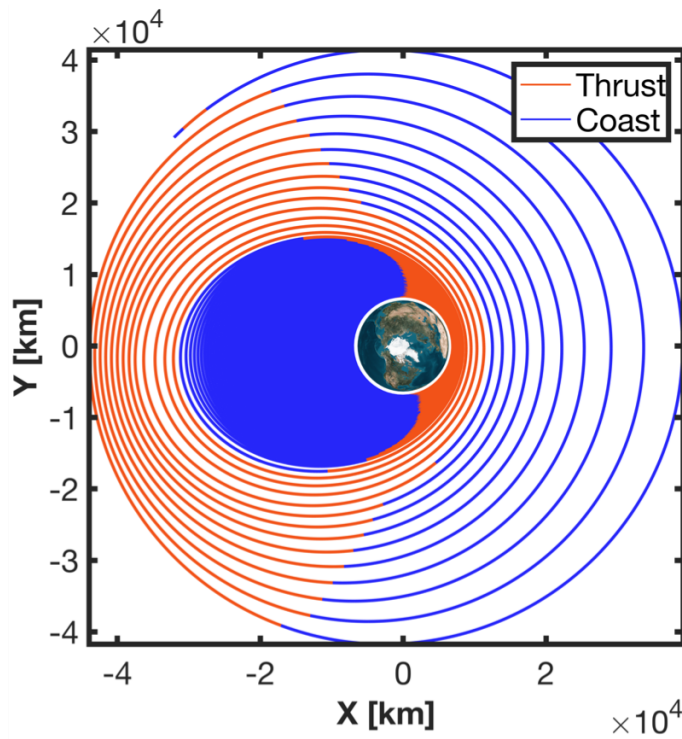
Mission Parameter	
m_0	300 (kg)
Thrust	1 (N)
I_{sp}	3100 (s)
μ_{Earth}	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
a_0	7000 (km)
e_0	0.01
i_0	0.05°
ω_0	0°
Ω_0	0°
θ_0	0°
a_{target}	42000 (km)
e_{target}	0.01
i_{target}	free
ω_{target}	free
Ω_{target}	free

Figure 3.2 shows the LEO-GEO transfer for the minimum-time case and when absolute effectivity coasting is included. When coasting is included, the effectivity checks limit the initial thrust to only occur near periapsis, therefore raising apoapsis. After growing apoapsis, the thrust centers around apoapsis to efficiently raise periapsis and reduce eccentricity. This process begins to resemble the low-thrust version of a Hohmann transfer and would become more exaggerated as η_a approaches 1.

Figures 3.3 and 3.4 show trajectories for the equatorial orbit to polar orbit transfer problem. These solutions demonstrate the influence that the Q-law gains can have on the resulting trajectory. Figure 3.3 was generated with $W_a, W_e, W_i = 1$. In this case, each element is weighted the same, but the large inclination change is



(a) Unity gains solution and no coasting. Flight time = 15.1 days. Propellant used = 43 kg.



(b) Unity gains solution with $\eta_a = 0.7$. Flight time = 35.2 days. Propellant used = 38.1 kg.

Figure 3.2: LEO-GEO trajectories.

Table 3.2: Equatorial to polar problem specifications.

Mission Parameter	
m_0	300 (kg)
Thrust	1 (N)
I_{sp}	3100 (s)
μ_{Earth}	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
a_0	10000 (km)
e_0	0.01
i_0	$1\text{e-}3^\circ$
ω_0	0°
Ω_0	0°
θ_0	0°
a_{target}	10000 (km)
e_{target}	0.01
i_{target}	90°
ω_{target}	free
Ω_{target}	free

the dominating term in Q . As a result, Q-Law raises the orbital radius in an effort to make the plane change easier to achieve. This results in a deviation from the target semi-major axis and eccentricity but results in a faster inclination change. The orbit is then lowered and circularized as the inclination approaches the target.

By increasing the weights on semi-major axis and eccentricity, priority is given to those elements in the thrust vector calculation and the deviation away from the target values is minimized. As an example of this, Figure 3.4 was generated with $W_a, W_e = 50$, $W_i = 1$. The increased W_a, W_e makes raising the orbital radius for a faster plane change less effective at reducing Q than performing a less-efficient plane change near the target semi-major axis and eccentricity. This results in a trajectory between the two orbits that is almost circular and maintains a nearly constant semi-major axis throughout the duration of the transfer. In the context of this problem, this actually results in a much longer flight time but provides a strong

example of how much influence gain tuning can have on the resulting solution.

3.5 Fast Variable Targeting

As previously mentioned, Q-Law suffers from the inability to target a specific fast variable. A successful Q-Law transfer will place the spacecraft somewhere on the target orbit, but the mission designer will have no control over the arrival true anomaly. If a specific point on the orbit must be reached, the spacecraft could simply coast until that point is reached after converging onto the other elements. In the scenario when perturbations are present, Q-Law thrust can be applied for orbit maintenance while waiting to reach the target anomaly. Thus the goal of fast variable targeting is to modify the Q-Law algorithm such that the full six-state is targeted faster than the baseline “achieve orbit then wait” approach.

An attempt at including fast variable targeting within the Lyapunov function was made in Reference [38]. However, this approach resulted in inconsistent flight times and took much longer to converge than the baseline approach. Two other unsuccessful attempts at sixth-element targeting are presented here. The first involves adding a true anomaly term into Q that takes the same form as the other elements. The second attempt uses a change of variables to replace true anomaly with a slow-moving term related to mean anomaly.

To include true anomaly in Q , the summation in Equation 3.6 is extended to include θ . The first step to calculating $\dot{\theta}_{xx}$ is to find the max rate of change in thrust vector space. To accomplish this, the thrust vector is aligned to be along the

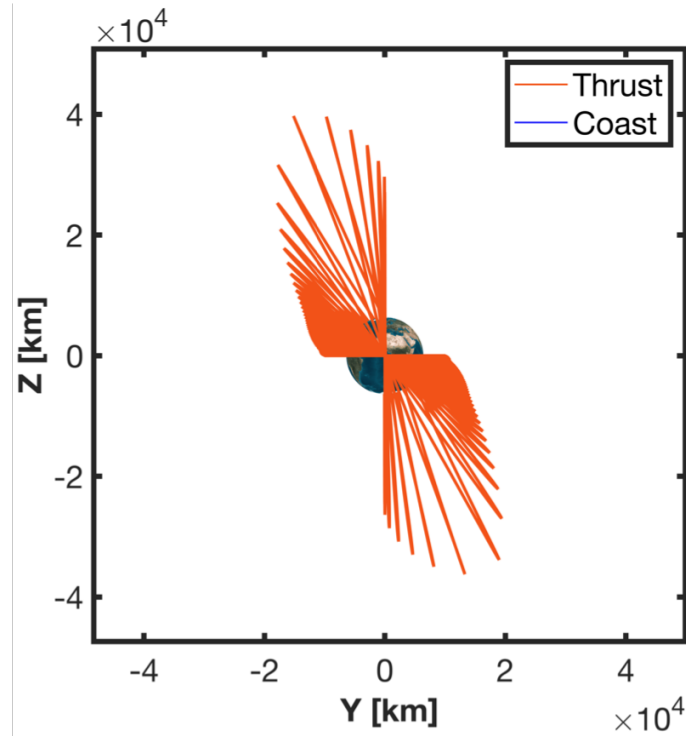
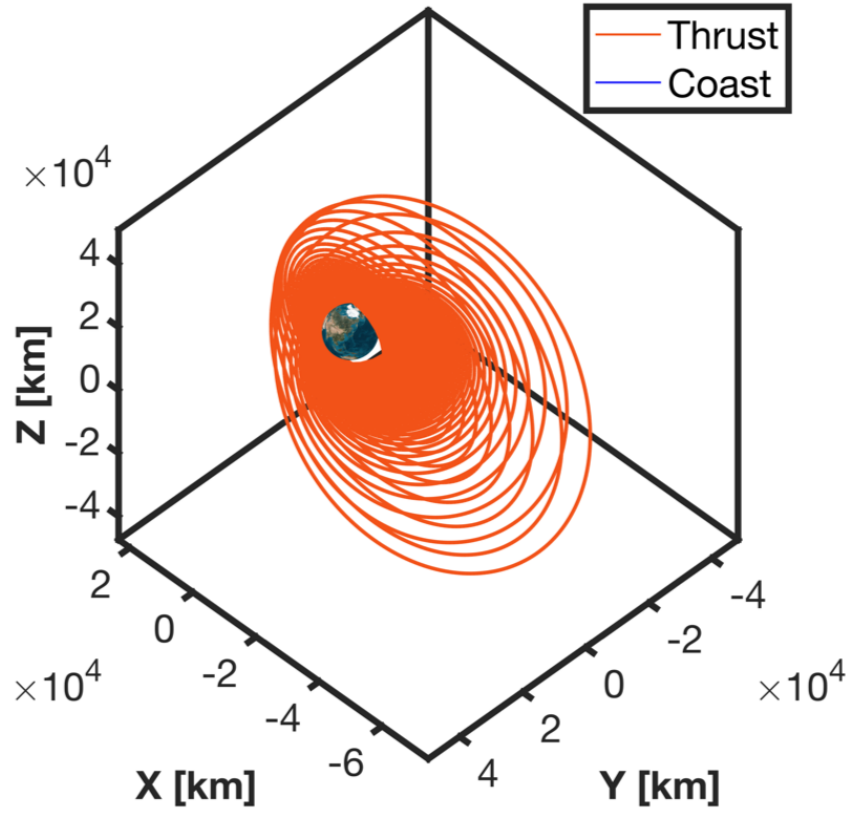


Figure 3.3: Equatorial to polar trajectory with unity gains and no coasting. Flight time = 33.3 days.

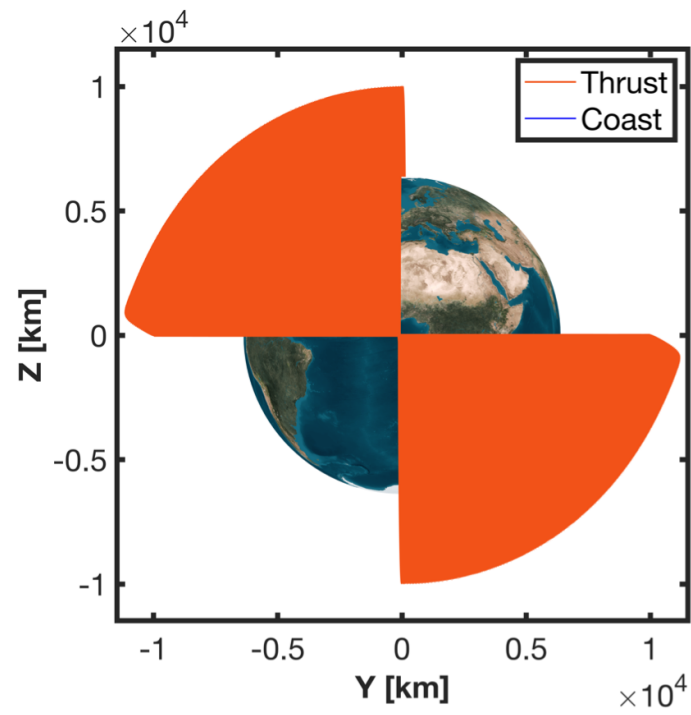
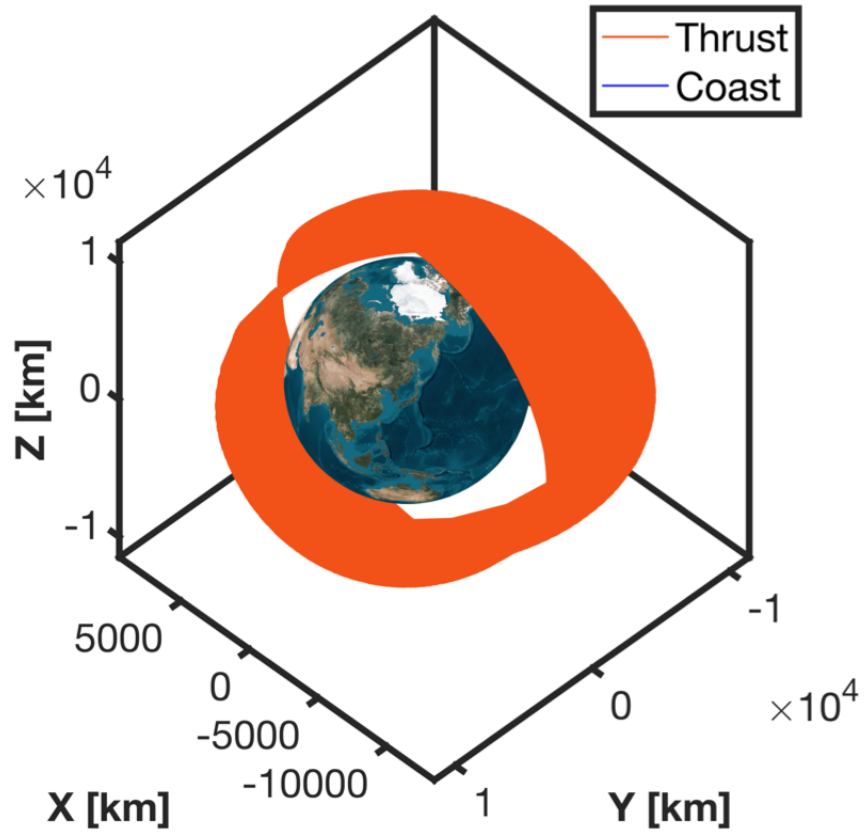


Figure 3.4: Equatorial to polar trajectory with weighted a, e and no coasting. Flight time = 42.3 days

direction formed by the coefficients from $\dot{\theta}$ in Equation 2.19:

$$\mathbf{u} = \frac{(\frac{p}{eh} \cos \theta, -\frac{p+r}{eh} \sin \theta, 0)^T}{\sqrt{(\frac{p}{eh} \cos \theta)^2 + (-\frac{p+r}{eh} \sin \theta)^2}} \quad (3.30)$$

When applied to $\dot{\theta}$, this results in

$$\dot{\theta}_{\mathbf{u}} = \frac{h}{r^2} + \frac{f}{eh} \sqrt{(p \cos \theta)^2 + ((p+r) \sin \theta)^2} \quad (3.31)$$

Then to find $\dot{\theta}_{xx}$, Equation 3.31 must be maximized over θ . Clearly true anomaly changes fastest at periapsis, $\theta = 0$, which leads to

$$\dot{\theta}_{xx} = \frac{h}{(a(1-e))^2} + \frac{pf}{eh} \quad (3.32)$$

This approach was tested on a simple eccentric transfer known as Case C in the original Q-Law manuscripts with a target true anomaly of apoapsis, $\theta = 180^\circ$. The problem parameters are listed in Table 3.3. Note that this problem has an unusually high thrust to mass ratio, so a shorter integration time step of 100 seconds was used. Additionally, the fixed step integration used to propagate the spacecraft can overstep the fast-moving true anomaly target, despite including a convergence tolerance. As a result, root-finding was performed at the target true anomaly on every revolution to check for element convergence at the proper point of the orbit.

As a baseline comparison, standard Q-Law was used to target the final orbit, and then the spacecraft waited until apoapsis was achieved. This solution trajectory is shown in Figure 3.5 and the corresponding osculating orbital elements are shown

Table 3.3: Eccentric orbit transfer problem specifications.

Mission Parameter	
m_0	300 (kg)
Thrust	9.3 (N)
I_{sp}	3100 (s)
μ_{Earth}	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
a_0	9222.7 (km)
e_0	0.2
i_0	0.573°
ω_0	0°
Ω_0	0°
θ_0	0°
a_{target}	30000 (km)
e_{target}	0.7
i_{target}	free
ω_{target}	free
Ω_{target}	free
θ_{target}	180°
W_a	1
W_e	1

in Figure 3.6.

The $W_\theta > 0$ cases yielded mixed results depending on the chosen values for W_θ . When a small value is used, $W_\theta < 0.3$, the solution remains unchanged from the baseline case as the low gain value results in minimal emphasis within the thrust vector calculation. However, larger values actually result in significant thrusting near the end of the transfer in an effort to reach apoapsis more quickly. This thrusting reduces the orbital radius, therefore speeding up $\dot{\theta}$, but typically pushes semi-major axis outside of the convergence tolerance. This results in a “miss” of the target elements despite reaching the target true anomaly more quickly, therefore adding another revolution onto the transfer. An exaggerated example of this behavior can be seen in the osculating elements of the $W_\theta = 10$ solution, shown in Figure 3.7. This

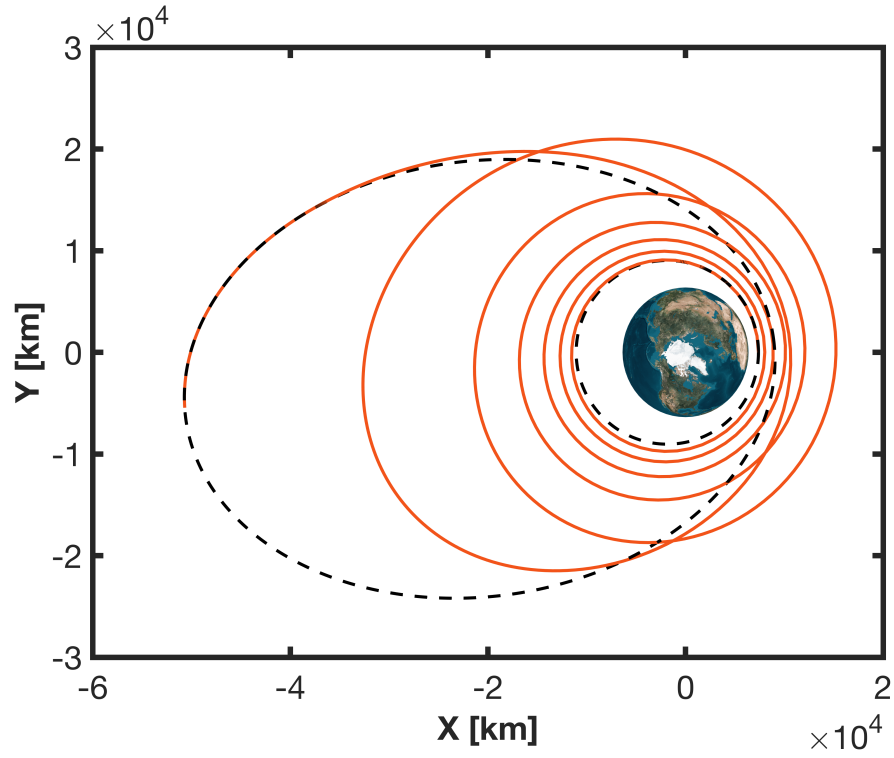


Figure 3.5: Eccentric transfer trajectory with initial and final orbits included.

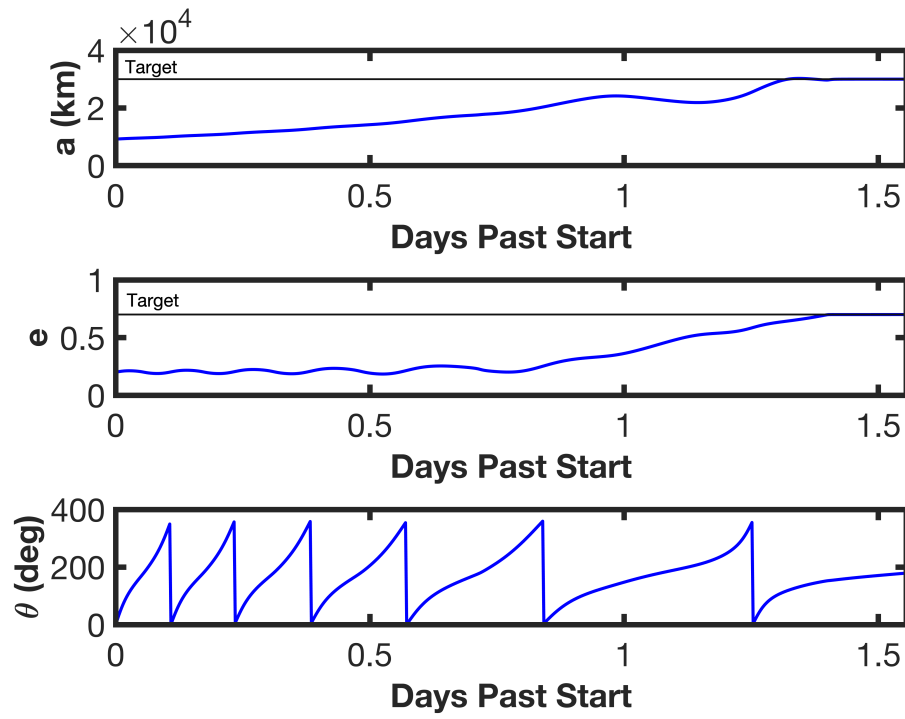


Figure 3.6: Baseline solution osculating orbital elements.

process repeats on every revolution, therefore prohibiting convergence, as shown in the plots of the osculating Q for the two cases, $W_\theta = 0$ and $W_\theta = 1$. Note that in the $W_\theta = 1$ and $W_\theta = 10$ cases, integration was terminated after 5 days to prevent the spacecraft mass from dropping too low.

It is noted that this approach resulted in thrusting to more quickly reach the target true anomaly, which would achieve the goal of fast variable targeting if semi-major axis were not pushed outside of the convergence region. If large semi-major axis tolerances are acceptable, then this approach could result in faster flight times. However, in a realistic mission design scenario, tight tolerances (< 50 km) are typically used and would not work well with the θ targeting thrust.

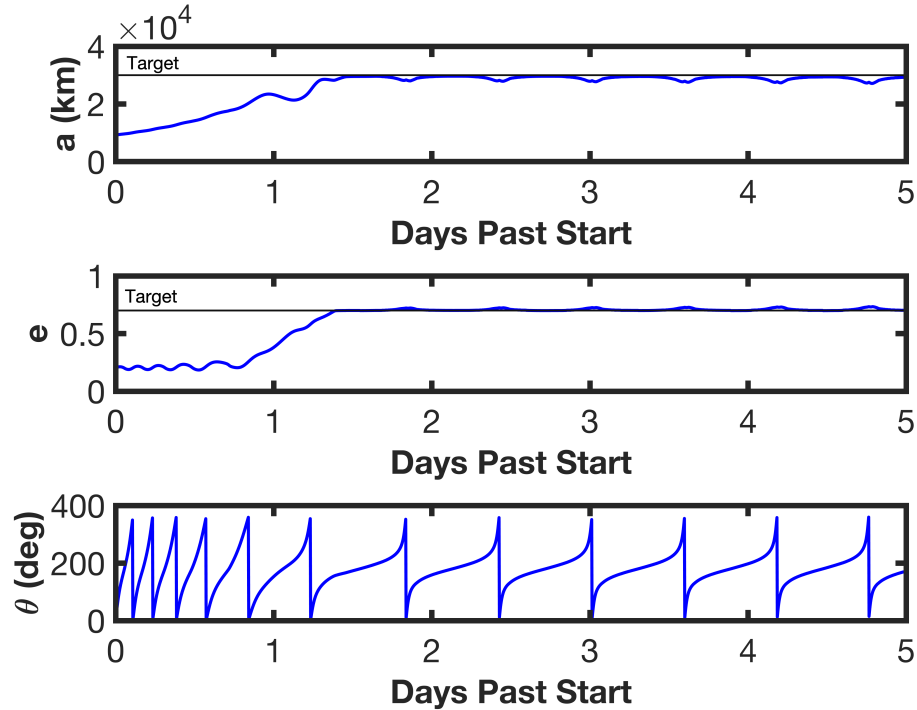


Figure 3.7: $W_\theta = 10$ solution osculating orbital elements.

The second 6th-element targeting effort replaced true anomaly with a slow

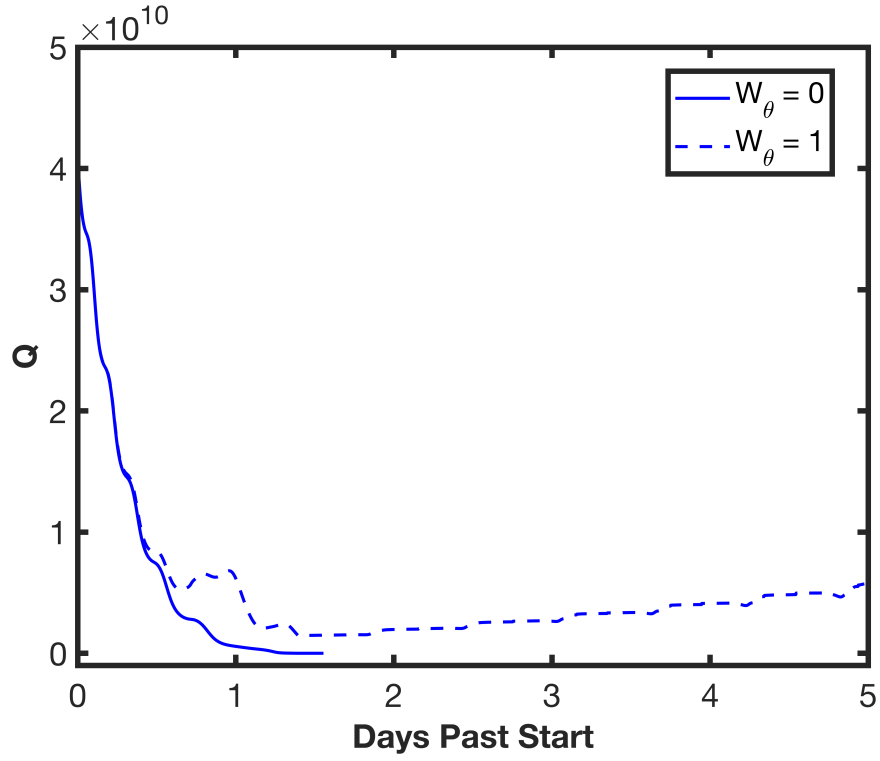


Figure 3.8: Osculating Q values.

varying term based on mean anomaly. Mean anomaly is defined as

$$M = \overline{M}_0 + \int_{t_0}^t n dt \quad (3.33)$$

where t_0 is a given epoch and n is the mean motion, defined as the $n = \sqrt{\frac{\mu}{a^3}}$.

Differentiating Equation 3.33 yields

$$\frac{d}{dt}M = \frac{d}{dt}\overline{M}_0 + n \quad (3.34)$$

Derivations of $\frac{d}{dt}\overline{M}_0$ are readily available in astrodynamics textbooks like References [25, 39], resulting in

$$\frac{d}{dt}\overline{M}_0 = \frac{d}{dt}M - n = \frac{b}{ahe}[(p \cos \theta - 2re)f_r - (p + r) \sin \theta f_\theta] \quad (3.35)$$

where b is the orbit semi-minor axis. The quantity \overline{M}_0 was used as the 6th element targeting variable, and $\frac{d}{dt}\overline{M}_0$ was included into the system dynamics. This change of variables was used in Reference [40] to study the averaged dynamics of a particle in orbit around Saturn.

The initial thought was that the slow variations of \overline{M}_0 would yield better targeting conditions than its oscillating, fast-moving anomaly counterparts. For use in Q-Law, $\dot{\overline{M}}_{0xx}$ is needed and was found using the same process that was used to find $\dot{\theta}_{xx}$ earlier in this section.

$$\dot{\overline{M}}_{0u} = \frac{fb}{ahe} \sqrt{(p \cos \theta - 2re)^2 + ((p + r) \sin \theta)^2} \quad (3.36)$$

Due to the complexity of this expression in terms of θ , the equation was discretely sampled for θ values between $[0, 2\pi]$ to identify $\dot{\overline{M}}_{0xx}$. This approach allowed for the targeting of a specific \overline{M}_0 as shown in Figure 3.9, but unfortunately did not yield proper mean anomaly targeting. The issue lies in that the semi-major axis will generally change during a low-thrust transfer, which yields changes in mean motion. As a result, $\int_{t_0}^t n dt$ is unknown at the arrival of the orbit, so proper targeting of \overline{M}_0 does not result in proper targeting of M . Further inspection of Reference [40] shows that the averaged perturbation dynamics studied did not result in semi-major axis variations, which allowed \overline{M}_0 to sufficiently model the particle's 6th element. Unfortunately this is generally not the case for low-thrust transfers.

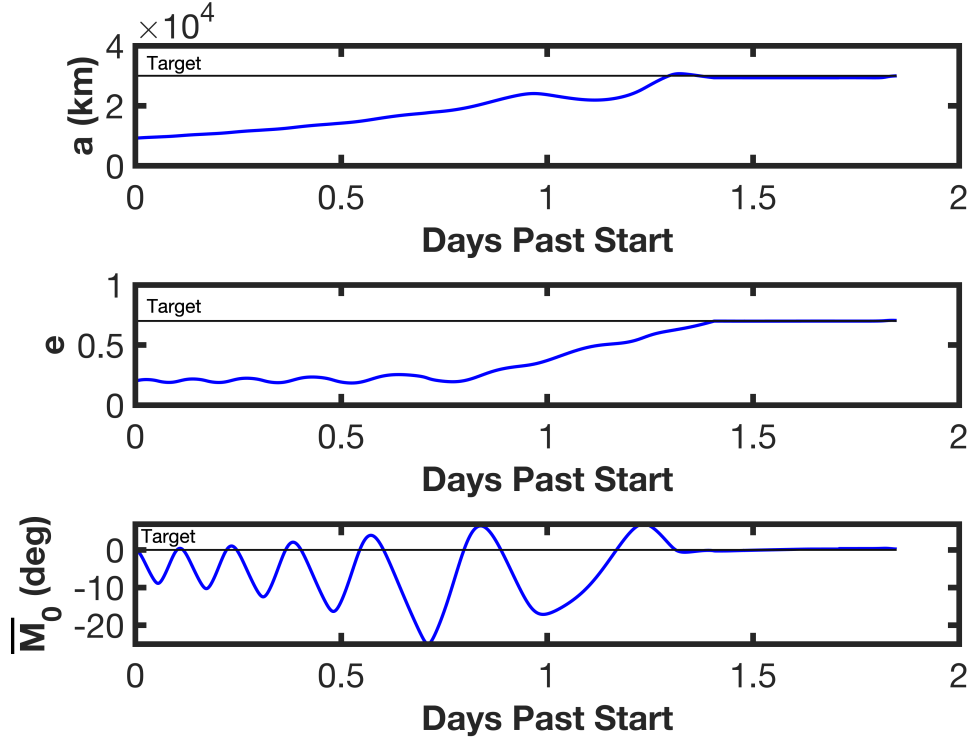


Figure 3.9: $\overline{M}_0 = 0$ targeting solution osculating orbital elements.

An iterative approach was attempted next, with the initial iteration proceeding without 6th element targeting. This provided a reference quantity for $\int_{t_0}^t n dt$ and an initial guess for the \overline{M}_0 target. Ideally, targeting this initial guess would result in a transfer that approaches the target orbit location. The target \overline{M}_0 can then be updated to account for the targeting errors until convergence on a transfer that hits the proper mean anomaly. However, including \overline{M}_0 targeting in the thrust vector calculation changes the thrust profile, therefore changing the element osculation and flight time. This results in significant changes in $\int_{t_0}^t n dt$. This behavior prohibits the iteration process from converging onto an \overline{M}_0 value that properly targets apoapsis. Results of the iterations are shown in Table 3.4.

The inability for Q-Law to effectively target all six elements poses a signifi-

Table 3.4: Results for iterative \overline{M}_0 targeting.

Iteration #	Flight Time (days)	Target \overline{M}_0 (deg)	$M - \int_{t_0}^t n dt$ (deg)
0	1.46	-	66.80
1	2.99	66.80	-142.86
2	2.90	-142.86	19.35
3	2.43	19.35	178.65
4*	5.00	178.65	-

* Did not converge within flight time limits.

cant problem for more complex applications like low-thrust rendezvous or intercept where hitting a time-varying true anomaly is required. To overcome this challenge, backward propagation can be used to design trajectories that target a fully defined state. This process is outlined in Chapter 5 and is used to design spiral Lunar transfers and Lunar gravity assist trajectories.

Chapter 4: Direct Collocation with a Q-Law Initial Guess

This chapter demonstrates the benefits of pairing Q-Law with a direct collocation approach for the many-revolution orbit transfer problem and is based on the work published in Reference [29]. An introduction to the collocation transcription method and nonlinear programming is also given. Here, Q-Law was used to generate a close-to-optimal trajectory for a GTO to GEO transfer with the inclusion of spherical harmonic gravity perturbations and eclipsing. This trajectory solution was then divided into two segments at a patch point, with the phase that falls after the patch point used as the initial guess for direct optimization. The fraction of Q-Law’s solution not used in the optimizer guess was unchanged, and the patch point was used to enforce continuous states between the Q-Law final condition and optimizer initial condition. The result was a continuous trajectory whose first phase was produced by Q-Law and second phase from the optimizer, as depicted in Figure 4.1. In this study, the fraction of the Q-Law solution used to seed the initial guess was varied, and the optimizer then produced the trajectory to the desired ending conditions. To solve the collocation phases, the GPOPS-II software was utilized, which uses an hp-adaptive Legendre–Gauss–Radau collocation method [41] and transcribes the optimal control problem into a large, sparse nonlinear program-

ming problem. Mesh-refinements were applied and the problem was re-optimized until satisfactory tolerances in the implicit integration were achieved [42].

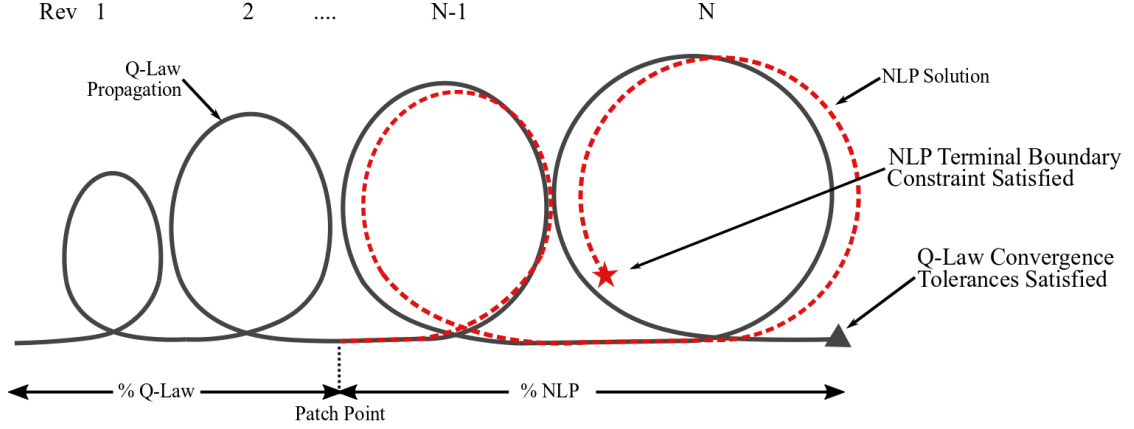


Figure 4.1: Q-Law and NLP problem setup.

Collocation is a common approach for solving the low-thrust transfer problem[19, 20, 43, 44, 45]. One challenge for planetocentric low-thrust trajectory optimization is the power discontinuities that result when the spacecraft passes through the central body shadow. John Betts and Graham and Rao both make notable efforts to include eclipsing within a collocation trajectory optimization scheme [19, 20]. Betts used a receding horizon algorithm and a Lyapunov control law different than Q-Law [46] to generate an initial guess for the trajectory. The trajectory was divided explicitly into thrust and coast phases, and the receding horizon algorithm determined an initial guess for how many phases are needed to achieve the target orbit. Graham and Rao solved the same minimum time GTO-GEO problem that is the focus of this chapter. They solved the problem initially without considering eclipses. The solution was then evaluated for eclipses and the solution terminated at first shadow entry. The non-eclipsed, minimum-time problem was solved again starting from the

shadow exit point, and this process was repeated until all eclipses were found and the target orbit was met. This approach results in a suboptimal final solution because the trajectory is piecewise optimized. In contrast to Betts and Graham and Rao, this research implemented a logistic eclipsing function [21] in the continuous dynamics to model the power/thrust drop inside the shadow. This allows the entire collocation problem to be reduced to a single phase.

The goal of this work was to exploit Q-Law’s speed and close-to-optimal solutions paired with direct collocation’s ability to utilize the entire Q-Law solution (path, controls, eclipse regions) as an initial guess, satisfy desired endpoint constraints, and produce optimal trajectories. Here, the trade between computation time and solution optimality is presented as the patch point location is varied along the Q-Law solution for both time-optimal and mass-optimal trajectories. This methodology was compared against a well-known GTO-GEO transfer problem that has been solved by several authors in previous literature [8, 9, 20].

4.1 Direct Collocation

The general trajectory design optimal control problem is defined as follows. Find the state, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$, control $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, initial time t_0 , and final time t_f on the interval $t \in [t_0, t_f]$ that minimizes the Bolza form cost function J in Equation 4.1 subject to the variable bounds listed in Equations 4.2 and 4.3 and the dynamic, boundary, and path constraints listed in Equations 4.4 to 4.6, respectively.

$$J = \phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (4.1)$$

$$\mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u \quad (4.2)$$

$$\mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u \quad (4.3)$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (4.4)$$

$$\mathbf{b}_l \leq \mathbf{b}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) \leq \mathbf{b}_u \quad (4.5)$$

$$\mathbf{g}_l \leq \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{g}_u \quad (4.6)$$

In this work, this continuous optimal control problem was solved using direct methods, where the optimal control problem is discretized using collocation so it can be formulated as a finite-dimension parameter optimization problem. Collocation is a parameterization method where the state and controls are approximated by piecewise polynomials, and derivative or integral dynamic constraints are enforced at discrete points in time along the trajectory. For a problem on the interval $[t_0, t_f]$, a mesh of size N will consist of discretized time points and the state and control parameters associated with each time point.

$$\Pi : t_0 = t_1 < \dots < t_N = t_f \quad (4.7)$$

$$\mathbf{Z} = [\mathbf{x}_1, \mathbf{u}_1, \mathbf{x}_2, \mathbf{u}_2 \dots \mathbf{x}_N, \mathbf{u}_N]^T \quad (4.8)$$

For numerical convenience, the optimal control problem time interval is mapped from $t \in [t_0, t_f]$ to $\tau \in [-1, +1]$ through the following transformation,

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0} \quad (4.9)$$

As a result of the new independent variable τ , the new transformed optimal control problem is defined as: Find the state, $\mathbf{x}(\tau) \in \mathbb{R}^{n_x}$, control $\mathbf{u}(\tau) \in \mathbb{R}^{n_u}$, initial time t_0 , and final time t_f that minimizes the Bolza form cost function J subject to the variable bounds listed in Equations 4.11 and 4.12 and the dynamic, boundary, and path constraints listed in Equations 4.13 to 4.15, respectively.

$$J = \phi(\mathbf{x}(-1), t_0, \mathbf{x}(+1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{+1} \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) d\tau \quad (4.10)$$

$$\mathbf{x}_l \leq \mathbf{x}(\tau) \leq \mathbf{x}_u \quad (4.11)$$

$$\mathbf{u}_l \leq \mathbf{u}(\tau) \leq \mathbf{u}_u \quad (4.12)$$

$$\frac{d\mathbf{x}(\tau)}{d\tau} = \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \quad (4.13)$$

$$\mathbf{b}_l \leq \mathbf{b}(\mathbf{x}(-1), t_0, \mathbf{x}(+1), t_f) \leq \mathbf{b}_u \quad (4.14)$$

$$\mathbf{g}_l \leq \mathbf{g}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \leq \mathbf{g}_u \quad (4.15)$$

There are many types of collocation schemes with varying degrees of fidelity, and the quality of the solution depends on the accuracy of the differential equation approximation. Common low-order schemes leverage Euler and trapezoidal integration steps, as shown in Equations 4.16 and 4.17, respectively.

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta\tau_i \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \tau_i; t_0, t_f) \quad (4.16)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\Delta\tau_i}{2} (\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \tau_i; t_0, t_f) + \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{u}_{i+1}, \tau_{i+1}; t_0, t_f)) \quad (4.17)$$

Defect constraints are evaluated at each mesh point and are defined to be the error between the predicted quadrature state and the actual state. The defect constraints for the two methods described above are defined as

$$\Delta_i = \mathbf{x}_i - \mathbf{x}_{i+1} + \Delta\tau_i \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \tau_i; t_0, t_f) \quad (4.18)$$

$$\Delta_i = \mathbf{x}_i - \mathbf{x}_{i+1} + \frac{\Delta\tau_i}{2} (\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \tau_i; t_0, t_f) + \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{u}_{i+1}, \tau_{i+1}; t_0, t_f)) \quad (4.19)$$

These two methods suffer from low-order accuracy, which limits their usefulness for complex problems. The Euler scheme is a first order method, $\mathcal{O}(h)$, and the trapezoidal method is a second order method, $\mathcal{O}(h^2)$. The fidelity of the solutions produced by these methods can be improved through a denser mesh that provides a better approximation to the system dynamics. However, increasing the number of segments will increase the complexity and computation expense of the resulting optimization problem. For real-world problems, it is usually more practical to use a higher order method that will require fewer segments to accurately model the system.

Higher order accuracy can be achieved by approximating the state and control with Lagrange polynomials and placing the nodes used to form these polynomials at the roots of a Legendre polynomial. This approach is used in the collocation scheme leveraged for this research, which is a variable-order Legendre-Gauss-Radau (LGR) method provided through the GPOPS-II optimal control software [41]. In this method, a particular mesh segment is approximated as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{X}_j L_j(\tau), \quad L_j(\tau) = \prod_{l=1, l \neq j}^{N_k+1} \frac{\tau - \tau_l}{\tau_j - \tau_l} \quad (4.20)$$

where $L_j(\tau)$, ($j = 1, \dots, N_k + 1$) is a basis of Lagrange polynomials and $[\tau_1, \dots, \tau_{N_k}]$

are the LGR collocation points [47]. The derivative of $\mathbf{X}(\tau)$ is found to be

$$\frac{d\mathbf{X}(\tau)}{d\tau} = \sum_{j=1}^{N_k+1} \mathbf{X}_j \frac{dL_j(\tau)}{d\tau} \quad (4.21)$$

The defect constraints are formulated by collocating the dynamic constraints in Equation 4.13 at the LGR points.

$$\Delta_i = \sum_{j=1}^{N_k+1} D_{ij} \mathbf{X}_j - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_0, t_f) = \mathbf{0} \quad (4.22)$$

where \mathbf{U}_i is the control approximation in the current mesh interval and

$$D_{ij} = \frac{dL_j(\tau)}{d\tau}, \quad i = 1, \dots, N_k, \quad j = 1, \dots, N_k + 1 \quad (4.23)$$

is the $N_k \times (N_k + 1)$ Legendre-Gauss-Radau differentiation matrix [48]. The cost function in Equation 4.10 is then approximated as

$$J \approx \mathcal{J} = \phi(\mathbf{X}_1^1, t_0, \mathbf{X}_{N_k+1}^N, t_f) + \sum_{n=1}^N \sum_{j=1}^{N_k} \frac{t_f - t_0}{2} w_i^n \mathcal{L}(\mathbf{X}_i^n, \mathbf{U}_i^n, \tau_i^n; t_0, t_f) d\tau \quad (4.24)$$

where N is the number of mesh segments in the trajectory, w_i^n are the LGR quadrature weights in mesh interval n , \mathbf{X}_1^1 is the approximation of $\mathbf{x}(-1)$, and $\mathbf{X}_{N_k+1}^N$ is the approximation of $\mathbf{x}(+1)$. This process is used to discretize the continuous optimal control problem into a parameter optimization problem known as a Non-linear Programming Problem. After a solution is found, the state and controls are interpolated to evaluate the constraint errors throughout the trajectory. Segments

with errors larger than the specified tolerances can be further divided into smaller segments or approximated with a higher order polynomial to reduce the error [42].

4.2 Nonlinear Programming

After the continuous optimal control problem is transcribed, it takes the form of a Nonlinear Programming Problem. A Nonlinear Programming Problem is an optimization problem where the minimization of an objective function is achieved by varying a set of decision variables \mathbf{x} , subject to a set of equality and inequality constraints. These problems take the following form:

$$\begin{aligned} \min_{\mathbf{x}} J(\mathbf{x}) \quad & \mathbb{R}^n \mapsto \mathbb{R} \\ s.t. \quad & l_b \leq \begin{pmatrix} \mathbf{x} \\ \mathbf{c}(\mathbf{x}) \end{pmatrix} \leq l_u \\ & \mathbf{x} \in \Omega \subseteq \mathbb{R}^n, \quad \mathbf{c} \in \mathbb{R}^m, \quad l_b, l_u \in \mathbb{R}^{n+m} \end{aligned} \tag{4.25}$$

where $J(\mathbf{x})$ is a scalar objective function, $\mathbf{c}(\mathbf{x})$ is a vector of constraint functions, $\{l_b, l_u\}$ are vectors of constant lower and upper bounds on \mathbf{x} and $\mathbf{c}(\mathbf{x})$, and Ω is the feasible region for the NLP. Feasible points for the problem posed in Equation 4.25 are those that satisfy all of the problem constraints. Gradient-based approaches, including sequential quadratic programming (SQP) and interior-point methods, are the most popular for solving this problem type. SNOPT and IPOPT are well-known

NLP software that leverage SQP and interior-point methods, respectively [49, 50].

NLP solvers attempt to solve the problem defined in Equation 4.25 by identifying a solution that satisfies the Karush-Kuhn-Tucker first order necessary conditions [51, 52]. To achieve this, the Lagrangian is defined to be

$$L(\mathbf{x}, \boldsymbol{\lambda}) = J(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x}) \quad (4.26)$$

The partial derivatives of the Lagrangian with respect to the decision variables \mathbf{x} and the Lagrange multipliers $\boldsymbol{\lambda}$ provide the first order necessary conditions for solving the NLP:

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \nabla_{\mathbf{x}} J(\mathbf{x}) - G^T(\mathbf{x}) \boldsymbol{\lambda} = 0 \quad (4.27)$$

$$\nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda}) = -\mathbf{c}(\mathbf{x}) = 0 \quad (4.28)$$

In Equation 4.27, $G(\mathbf{x})$ represents the Jacobian of the constraint functions. The accuracy of the dense Jacobian entries can have significant effects on the NLP solver's ability to converge on a solution satisfying Equations 4.27 and 4.28. Poor Jacobian accuracy can require additional iterations for the optimizer to converge or lead to numerical difficulties that prevent a solution from being found.

4.3 Numerical Differentiation Techniques

Solving an NLP requires calculating the Jacobian of the system constraints. If analytical expressions for these derivatives are known, it is very beneficial to supply them directly to the optimizer. However, analytical derivatives are often difficult to generate or not available for complex problems. In this scenario, it is common practice to estimate the Jacobian entries numerically, with the most popular technique being finite differencing. Most NLP solvers include finite differencing capabilities built into the software. The most commonly used finite differencing formulas are shown in Equations 4.29 and 4.30.

$$\frac{\partial f(x)}{\partial x} = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h) \quad (4.29)$$

$$\frac{\partial f(x)}{\partial x} = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2) \quad (4.30)$$

While finite differencing presents a straightforward and easy-to-implement method of derivative approximation, there are several drawbacks to this approach. First and foremost, the accuracy of the calculated derivative directly corresponds to the chosen step size h . As indicated in Equations 4.29 and 4.30, the Taylor series truncation error can be reduced by using a smaller step size, but only up to a certain point. Choosing a step size too small will increase the floating point round-off error on a computer, leading to derivative inaccuracies. Additionally, there can be significant computational cost associated with finite differencing. Even the first order

method shown in Equation 4.29 requires multiple function evaluations, and if the function is computationally expensive to evaluate, this can lead to significant runtime. This effect is amplified if higher-order approximations are used. As a result, it is often best to avoid finite-differencing when possible.

Another technique for numerically estimating derivatives is the complex-step method [53]. In this approach, a truncated Taylor series is also used. However, unlike in finite differencing, the perturbing step is chosen to be in the imaginary direction. Performing this truncation and isolating the first derivative term leads to

$$\frac{\partial f(x)}{\partial x} = \frac{\text{im}(f(x + ih))}{h} + \mathcal{O}(h^2) \quad (4.31)$$

In this expression, there is no subtraction term that could lead to floating point round-off, as was the case with finite differencing. As a result, the step size can be chosen to be very small, leading to a derivative estimation accuracy within floating point precision.

The last numerical differentiation technique used in the research is the dual number automatic differentiation approach. When using dual numbers, the element ϵ is defined to have the following property:

$$\epsilon^2 = 0 \quad (4.32)$$

Using this property yields the following relationship:

$$(a + b\epsilon)(c + d\epsilon) = ac + (ad + bc)\epsilon \quad (4.33)$$

Next, a Taylor series expansion with a perturbation in the dual direction is evaluated.

$$f(x + \epsilon) = f(x) + \epsilon f'(x) + \epsilon^2 \frac{f''(x)}{2} + \epsilon^3 \frac{f'''(x)}{6} + \dots \quad (4.34)$$

Applying the property described in Equation 4.32 to Equation 4.34 shows that the second order and higher terms vanish.

$$f(x + \epsilon) = f(x) + \epsilon f'(x) \quad (4.35)$$

Therefore, the solution from evaluating $f(x)$ with a unity dual part contains the exact functional derivative in its dual part. This capability has been implemented in the JuliaDiff library, which provides an efficient and easy-to-use means to calculate numerically exact derivatives [54].

4.4 Optimization Problem Setup

The goal of the single phase optimal control problem is to determine the trajectory $\mathbf{x}' = [p(L), f(L), g(L), h(L), k(L)]$ and the control variables \mathbf{u} containing the normalized thrust vector $\mathbf{T}_n = (u_r, u_\theta, u_h)$ that minimize the objective function J . Here, the objective function was TOF, as shown in Equation 4.36.

$$\min J = t_f \quad (4.36)$$

subjected to the dynamical constraints of Equations 2.24, 2.29 and 2.30. Additionally, there were several other constraints. Equation 4.37 arises because the

time-optimal problem was solved with thrust fixed at the maximum value, with the only exception being when the spacecraft was shaded in an eclipse.

$$u_r^2 + u_\theta^2 + u_h^2 = 1 \quad (4.37)$$

Equation 5.1 ensured a continuous transition between the Q-Law phase and the collocation phase, regardless of the selected transition point.

$$\begin{aligned} \mathbf{x}^{opt} \left(t_0^{opt} \right) - \mathbf{x}^{QLaw} \left(t_f^{QLaw} \right) &= 0 \\ t_0^{opt} - t_f^{QLaw} &= 0 \end{aligned} \quad (4.38)$$

Equation 5.2 enforced the desired terminal condition (i.e geosynchronous orbit).

$$\begin{aligned} p^{opt} \left(t_f^{opt} \right) &= p^{des} \\ f^{opt} \left(t_f^{opt} \right) &= f^{des} \\ g^{opt} \left(t_f^{opt} \right) &= g^{des} \\ h^{opt} \left(t_f^{opt} \right) &= h^{des} \\ k^{opt} \left(t_f^{opt} \right) &= k^{des} \end{aligned} \quad (4.39)$$

The dynamics were evaluated with L as the independent variable for numerical

efficiency, but the patch point location and continuity constraints were formulated with time as the independent variable. No specific fast variable was targeted in the terminal constraints. When eclipses were present in the collocation phase, the logistic function allowed for a single phase transfer from the transition point to the terminal condition. This means the dynamics, including control, were continuous without any discrete break points. Bounds were also placed on the state and control variables to ensure that f , g , h , k , u_r , u_θ , and u_h stay within ± 1 .

The equivalent mass-optimal control problem was also solved using the minimum-time Q-Law solution TOF as the constrained duration for GPOPS-II. In this formulation, thrust was allowed to vary between zero and T_{max} , thus allowing for non-eclipsed coast arcs. The optimization problem then became

$$\min J = -m_f \tag{4.40}$$

with the normalized thrust path constraint modified to

$$0 \leq u_r^2 + u_\theta^2 + u_h^2 \leq 1 \tag{4.41}$$

and the NLP solver determining both the control direction and magnitude.

4.5 Example: GTO to GEO Transfer

Q-Law was used to generate a solution to a well-known GTO-GEO low-thrust trajectory problem previously solved in the literature [8, 9, 20]. The Q-Law trajec-

tory propagation was executed inside Matlab’s built-in genetic algorithm parallelized across 4 processors [55, 56, 57]. A population size of 50 was used and the gain search was restricted between 1e-3 and 10 for the elements of interest, W_a , W_e , W_i . The algorithm used a crossover fraction of 0.8, migration fraction of 0.2, and a convergence tolerance of 1e-5. Setting W_Ω and W_ω to zero eliminates those elements from the thrust vector calculation, allowing Q-Law to focus on the elements of interest for the GTO-GEO transfer.

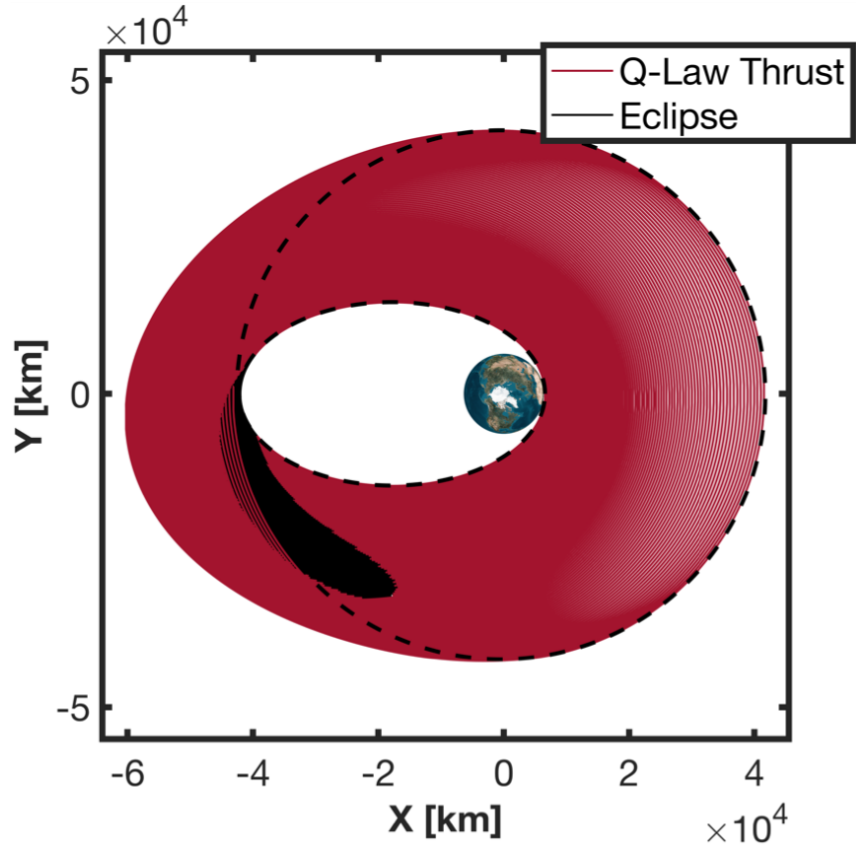
4.5.1 Time-Optimal Results

First, it is noted that the Q-Law controls rely on Gauss’s variational equations which have singularities at $e = 0$ and $i = 0$, so eccentricity and inclination just above zero were targeted for GEO. Additionally, Q-Law asymptotically approaches the desired orbital elements, so it was very beneficial to specify a convergence tolerance for each element. The tolerances allowed the Q-Law algorithm to terminate when the trajectory closely approached the desired orbit, but it resulted in a solution that did not exactly meet the endpoint conditions. However, allowing the NLP solver to optimize the last phase of the trajectory improved the solution by both improving the optimality and ensuring the boundary constraints were satisfied within the NLP’s tolerances, which are generally much tighter than Q-Law’s. The results are presented according to the percent of the solution produced by the optimizer in time (e.g. the 50% NLP case refers to the solution where the first 50% of the trajectory in time was produced by Q-Law and the last 50% was produced by the optimizer). For this

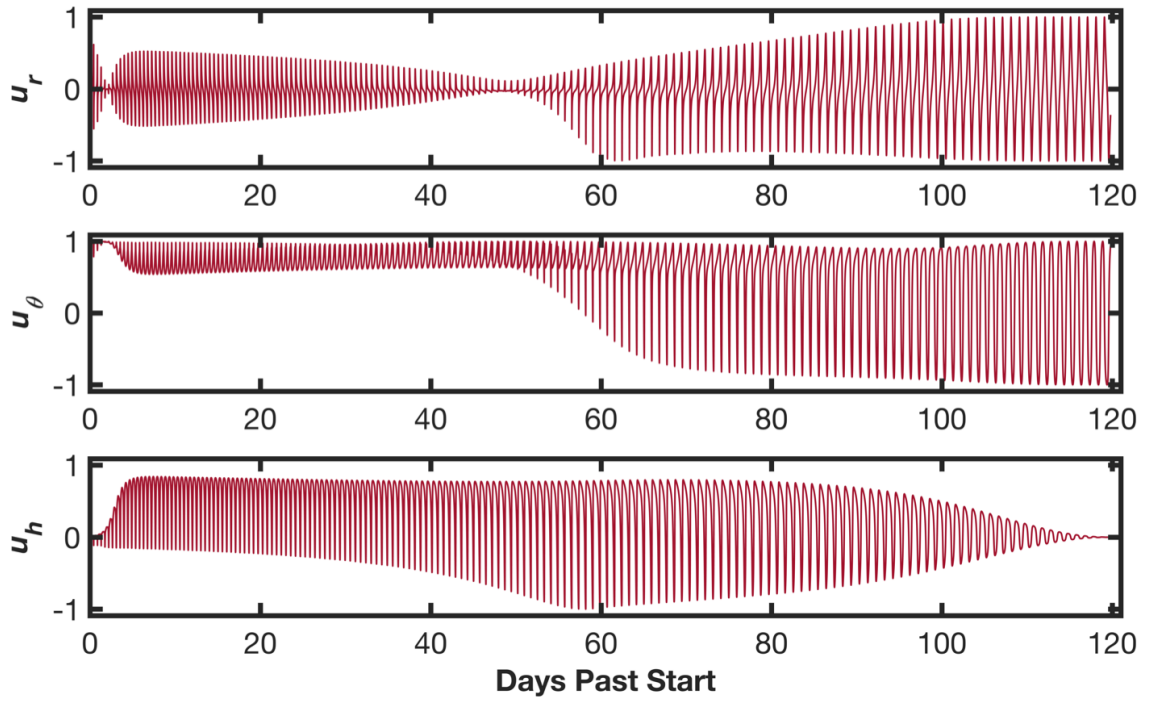
particular problem, the 50% point in time into the trajectory occurred very close to the final eclipse. The patch point was placed at the end of the last eclipse and used as the 50% patch point location to evaluate the computation time when no eclipses were present in the optimization phase. The Q-Law states were converted to MEE before use in GPOPS-II and were scaled using l_{scale} as a length scale, t_{scale} as a time scale, and L_{scale} as a fast variable scale. The values used for the time and fast variable scales naturally presented themselves for the chosen GTO-GEO problem, but the length scale proved to be more sensitive when eclipses were included. Some trial and error was needed to find a suitable value.

The GTO-GEO initial and target orbits are provided in Table 4.1. The GTO elements refer to an orbit with a perigee of 6563.6 km, an apogee of 42164.3 km, and an inclination of 28.5° . The GEO elements refer to a circular, equatorial orbit with a radius of 42165 km. Table 4.2 outlines the settings and parameters used for the dynamics models, Q-Law algorithm, and GPOPS-II. Computations were executed in Matlab on a Windows laptop with 16 GB RAM and Intel(R) Core(TM) i7-4712HQ CPU 2.3 GHz processor. Table 4.3 presents the time-optimal results, compared to previous literature. The 0% solution trajectory and control history are shown in Figures 4.2a and 4.2b. The 50% solution trajectory and control history are shown in Figures 4.3a and 4.3b. The 100% solution trajectory and control history are shown in Figures 4.4a and 4.4b, and the osculating orbital elements for all time-optimal cases are shown in Figures 4.5a to 4.5c.

Although Q-Law does not exactly meet the endpoint constraints, the NLP solver ensured that all constraints were met and increased optimality, as demon-

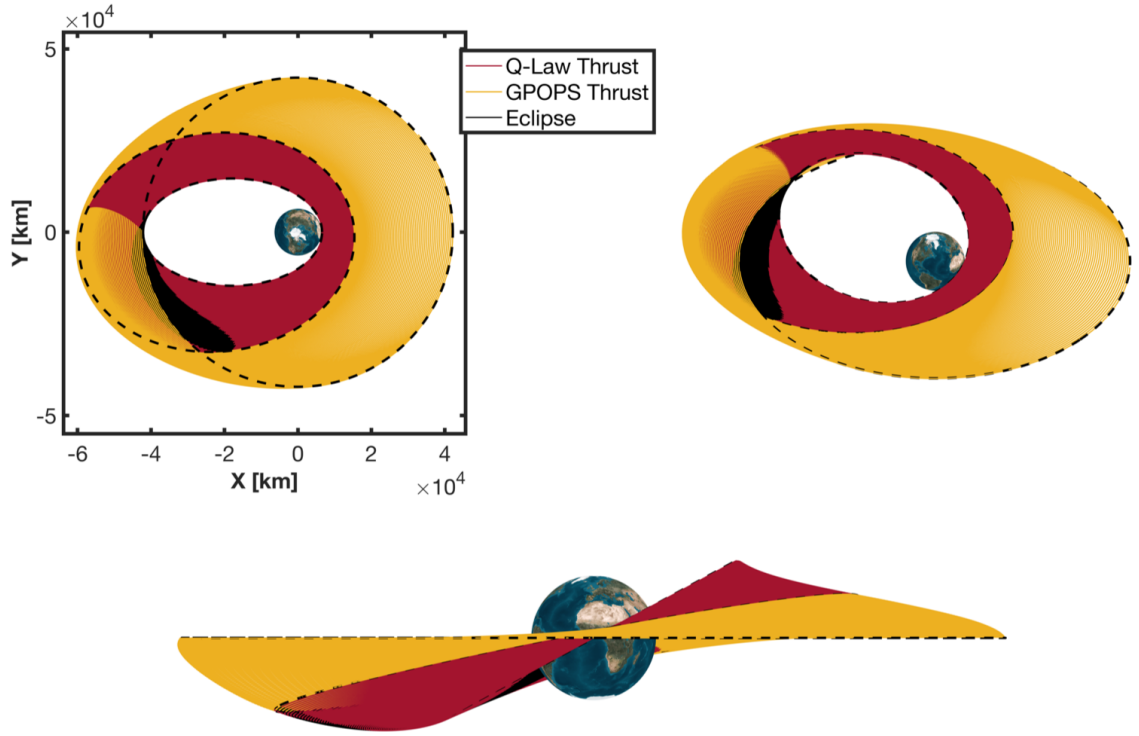


(a) Q-Law trajectory.

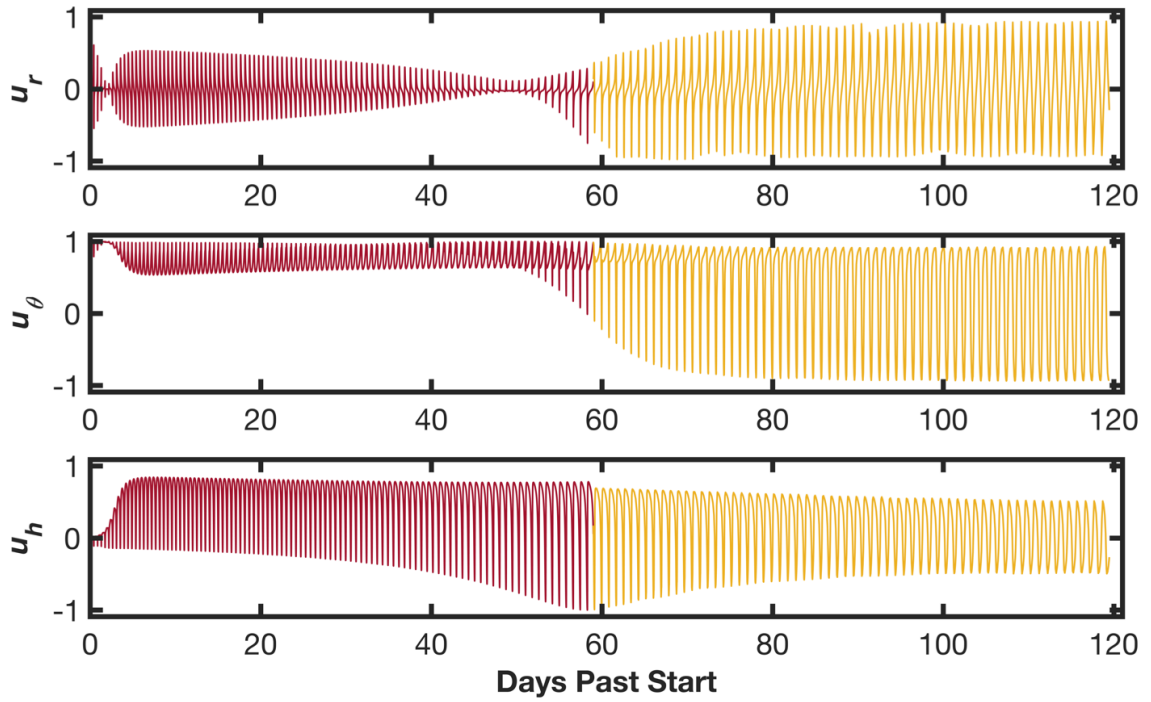


(b) Control history.

Figure 4.2: Q-Law solution from GTO-GEO.

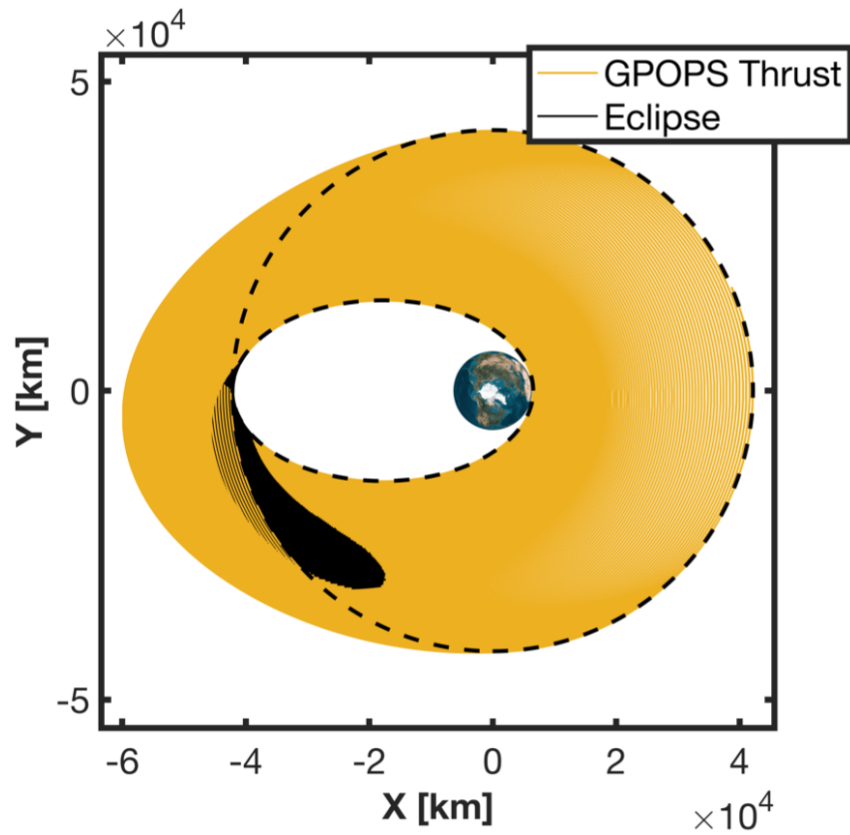


(a) Trajectory. Eclipsed portions of the trajectory are black. The top-left figure is an equatorial projection along the x - y plane, the top right figure is a 3-D view, and the bottom figure is an x - z projection.

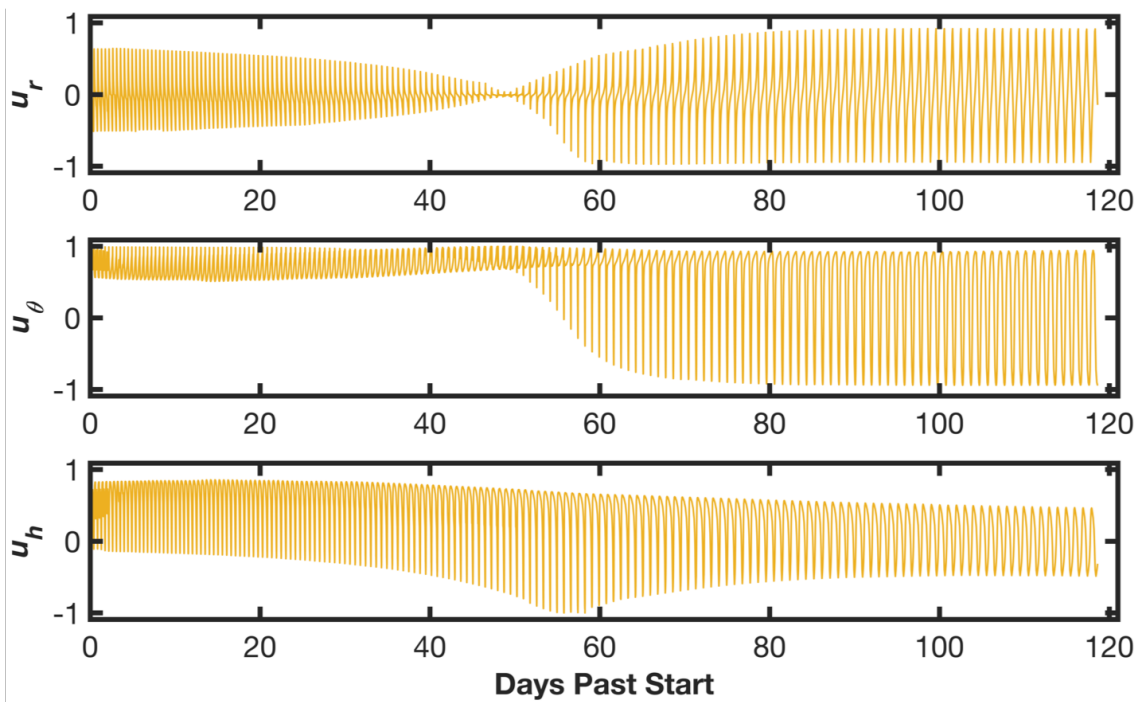


(b) Control history.

Figure 4.3: GTO-GEO solution for the 50% case. The Q-Law phase appears in red and the GPOPS-II phase in yellow.



(a) Optimal trajectory



(b) Optimal control history.

Figure 4.4: GPOPS-II solution from GTO-GEO.

Table 4.1: Initial and target orbits in equinoctial coordinates

Orbit	p (km)	f	g	h	k
GTO	11359.07	0.7306	0	0.25396	0
GEO	42165	0	0	0	0

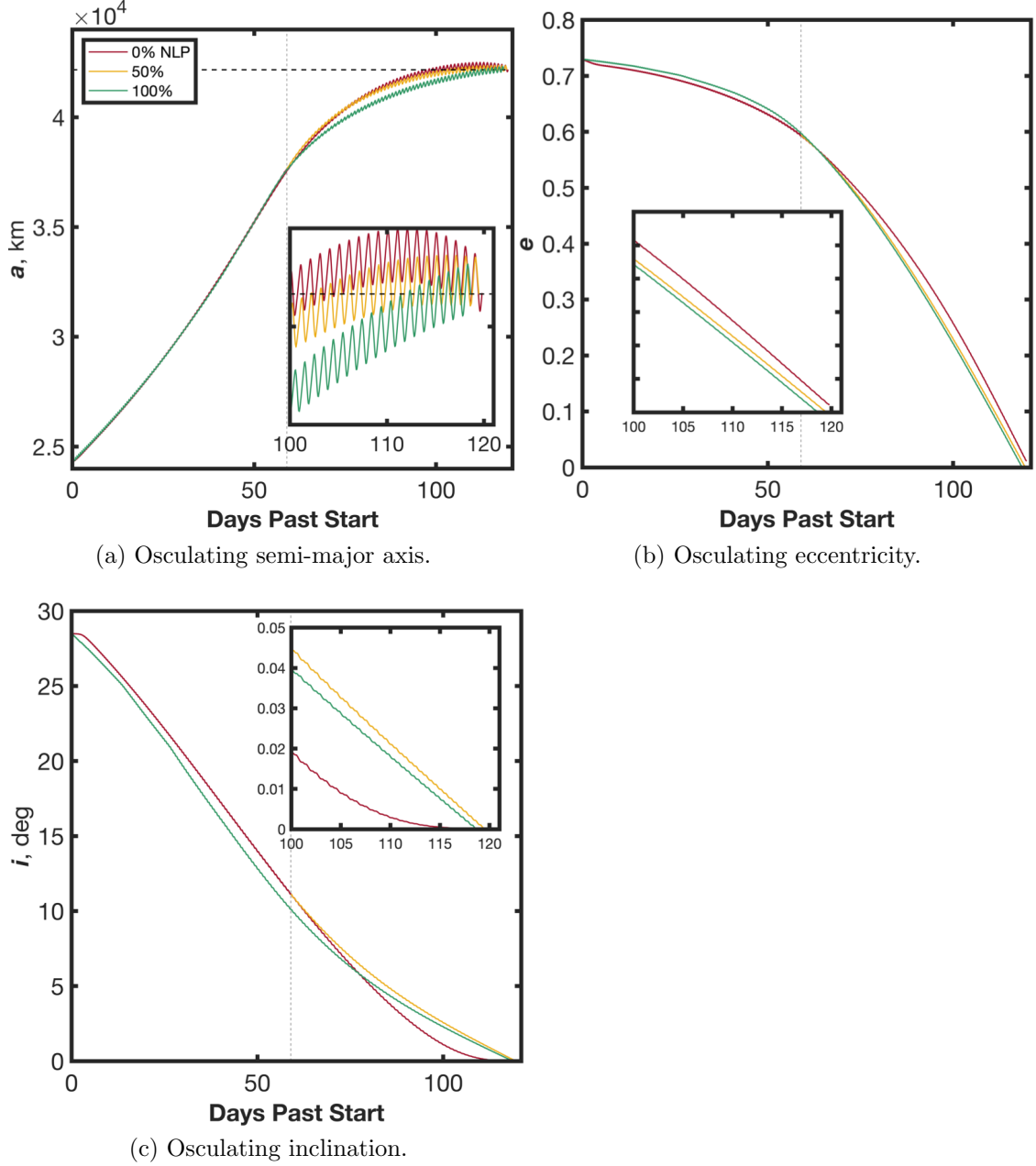


Figure 4.5: Time-optimal osculating orbital elements.

Table 4.2: Dynamics constants, Q-Law parameters, and GPOPS-II settings.

Constant	Value
Universal Time Departure	March 22, 2000 00:00:00.000
m_0	1200 kg
I_{sp}	1800 s
P	5 kW
λ	1×10^{-4}
η	0.55
g_0	$9.80665 \times 10^{-3} \text{ km/s}^2$
μ_{\odot}^*	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
R_{\odot}^*	6378.14 km
R_{\odot}	695500 km
J_2	1086.639×10^{-6}
Perturbing Bodies	None
Planetary Ephemeris Kernel	de430
Q-Law Parameter	Value
W_a	2.406
W_e	1.786
W_i	9.469
W_{Ω}	0
W_{ω}	0
η_a	0
η_r	0
a_{target}	42165
e_{target}	0.01
i_{target}	0.01°
a_{tol}	10 km
e_{tol}	0.001
i_{tol}	0.01°
GPOPS-II Setting	Value
NLP Solver	SNOPT
Derivative Type	Central Differences
Collocation Method	RPM-Differentiation
Scales Method	None
NLP Tolerance	1×10^{-6}
Mesh Method	hp-LiuRao[42]
Mesh Tolerance	1×10^{-5}
Max Col. Pts	14
Min Col. Pts	2
l_{scale}	$17 \times 10^3 \text{ km}$
t_{scale}	$100 \times 86400 \text{ s}$
L_{scale}	$50 \times 2\pi \text{ rads}$
m_{scale}	1000 kg
Mesh Pts Per Revolution (Non-Eclipsed)	3
Collocation Pts Per Mesh (Non-Eclipsed)	5
Mesh Pts Per Rev. (Eclipsed)	7
Col. Pts Per Mesh (Eclipsed)	5
c_s	289.78
c_t	1

strated by the 50% and 100% case results. Using their multi-phase method, Graham and Rao[20] found a solution with TOF= 121.22 days and $\Delta m = 172.23$ kg, which is 2.19% longer than the 100% NLP solution. When the patch point was placed in front of many Q-Law eclipses, which only occurred in the 100% NLP case, the optimization became more sensitive and sometimes got stuck in a local minimum near the Q-Law solution. To increase the likelihood of producing a more globally optimal result and overcome the eclipse induced sensitivity, the 50% NLP case was included in the guess for the 100% NLP case (i.e. the first half of the initial guess was produced by Q-Law and the second half by GPOPS-II). The improved solution optimality and robust convergence was well worth the relatively small computation time needed to produce the 50% NLP result used in the 100% NLP case initial guess. However, for a similar transfer with no/fewer eclipses, the 50% case would likely not need to be included in the initial guess for the 100% case. Additionally, the number of initial mesh points was increased to 7 for revolutions containing Q-Law eclipses. This method successfully produced a single-phase, 163-revolution trajectory with eclipsing from GTO to GEO, but it required significantly more computation time and mesh refinements than the 50% case. This is because additional mesh points were required to accurately model the transition phases associated with entering and exiting an eclipse, and eclipses were not present in the 50% case optimization phase. When the patch point was placed at the Q-Law trajectory start, the resulting TOF is within 0.3% of the averaged solution. The discrepancies likely result from the errors associated with approximating eclipses with a logistic function and getting stuck in a local minimum because of an imperfect initial guess. It is noted

that the computation time for the 0% case (Q-Law only) is the time for one Q-Law evaluation and not the time required by the genetic algorithm to identify optimal gains. Genetic algorithm computation time will vary with the computing resources available for parallelization.

Table 4.3: Time-optimal problem results for varying patch point location and literature comparison

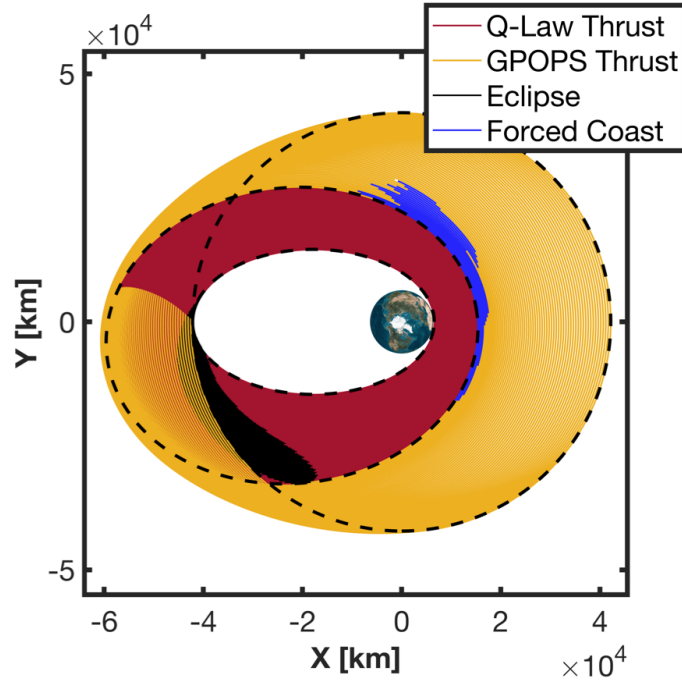
%NLP	%Q-Law	<i>TOF</i> (days)	Δm (kg)	CPU Time
0	100	119.79	170.81	< 1 min
50	50	119.46	170.30	4 min
100	0	118.62	169.44	2.75 hrs
Graham & Rao[20]	-	121.22	172.23	-
Orbital Averaging[8, 9]	-	118.35	169.2	-

Although this analysis was able to produce the 100% NLP case, the computation time needed to produce this solution was very large. This is partly because the partial derivatives for this problem were estimated using finite differencing. As previously discussed, this adds computation time and can require more iterations from the optimizer. The other source of computation time is the inclusion of the eclipses within the optimization. As more eclipses were introduced, more collocation points and mesh iterations were needed to accurately represent the thrust dynamics in the shadow transition region. It may be more desirable for a mission designer to use gain-tuned Q-Law (rather than NLP solvers) as a means to identify eclipses and quickly produce an initial solution that is close to optimal. Then, an optimization of the Q-Law trajectory starting just after the last eclipse will increase solution optimality and meet the endpoint constraints without adding significant computation time.

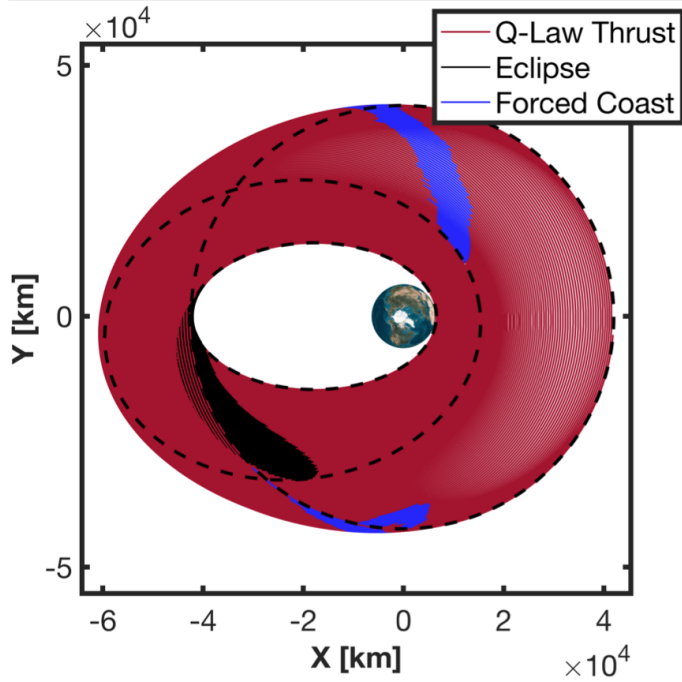
4.5.2 Mass-Optimal Results

The 50% mass-optimal problem was also investigated. Similar to the time-optimal problem previously discussed, a patch point was placed at the 50% point along the minimum-time Q-Law trajectory, with the phase after the patch point serving as an initial guess for GPOPS-II. In this analysis, the TOF is fixed as the Q-Law TOF and the thrust magnitude was allowed to vary between off and T_{max} , as described in Equation 4.41. The same constants, Q-Law parameters, and optimizer settings as the minimum-time problem were used. The only exception was that IPOPT with a tolerance of 1×10^{-6} was used instead of SNOPT to solve the NLP because IPOPT was found to be more efficient when thrust magnitude is allowed to vary. Q-Law’s effectivity checks were then used to investigate the additional TOF needed for Q-Law to match the propellant savings observed in the 50% mass-optimal solution. For direct comparison to the 50% mass-optimal case, effectivity checks are not used until after the 50% patch point. Each solution is described in Table 4.4 and the resulting trajectories and control shown in Figures 4.6a, 4.6b, and 4.7. Relative effectivity, as described in Equation 3.27, was used as the cutoff method. Note that Table 4.4 does not report values for W_Ω or W_ω as those elements are not targeted for this problem.

These results highlighted the deficiencies of the effectivity checks, as they need almost 3 days more flight time to match the GPOPS-II propellant savings and can be computationally expensive. In the mass-optimal problem, fixing the TOF saved more than 1 kg of propellant over the initial Q-Law solution. Although



(a) Mass-optimal solution for 50% case.



(b) Q-Law trajectory with effectivity checks to match GPOPS-II minimum-mass propellant savings.

Figure 4.6: Mass-optimal and Q-Law effectivity equatorial x - y trajectory projections for the GTO-GEO 50% Case. Optimal/forced coasts appear in blue.

Table 4.4: Mass-optimal results for the 50% case and Q-Law with effectivity coasting mechanism. The 50% time-optimal and original Q-Law solutions are also presented for comparison.

Problem	TOF (days)	Δm (kg)	η_r	CPU Time (min)
50% Mass-Optimal	119.79	169.19	-	30
Q-Law + Effectivity	122.44	169.19	0.0151	4
Q-Law	119.79	170.81	0	< 1
50% Time-Optimal	119.46	170.30	-	4

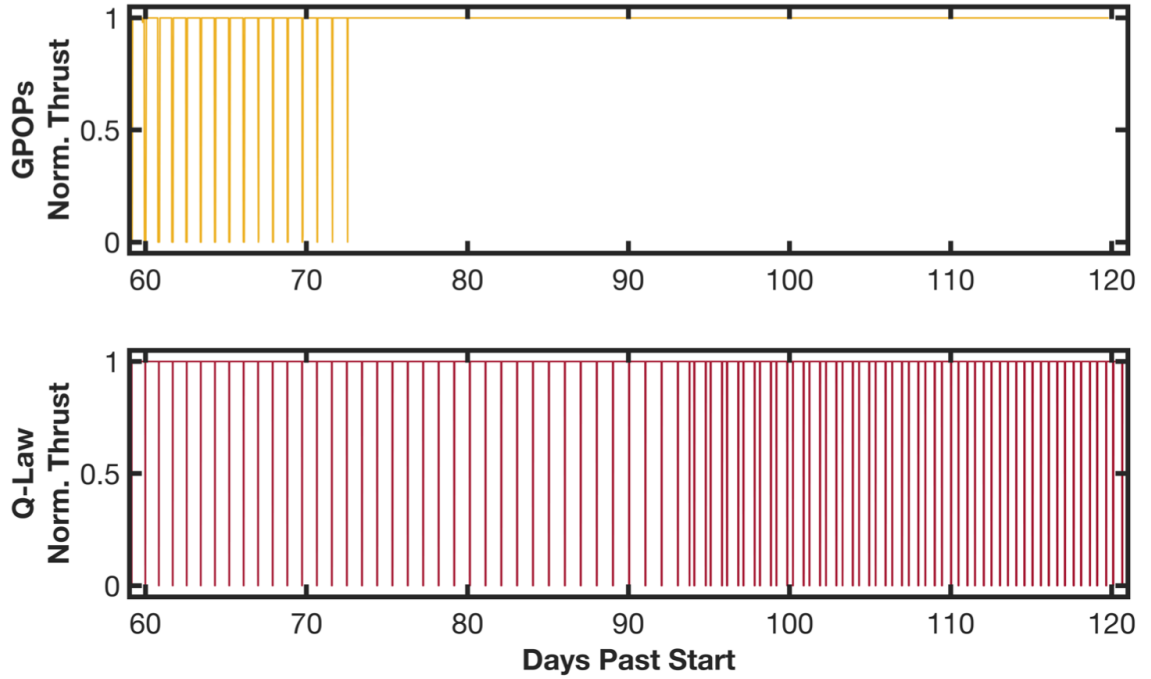


Figure 4.7: Normalized thrust magnitude for mass-optimal solution and Q-Law with effectivity checks. The first 60 days are omitted as they are the same for each case.

only small propellant savings were found here, this demonstrated the ability to solve mass-optimal problems using a minimum-time Q-Law guess, which may be relevant when a properly gain-tuned Q-Law successfully produces a minimum-time solution whose TOF meets mission requirements. Mission designers could then constrain the TOF and optimize over any fraction of the Q-Law trajectory to reduce propellant usage. The mass-optimal problem is generally more difficult to solve, and the optimized solution demonstrated bang-bang control in the early stages of the optimized transfer. Figure 4.6a shows that the optimal coast locations occur earlier along the trajectory near periapsis, and the coasting stops when additional coasts would violate the TOF constraint. Comparing Figures 4.6a and 4.6b, it can be seen that Q-Law’s relative effectivity check behaves differently than GPOPS-II and tends to coast near true anomalies $\pm\frac{\pi}{2}$. These coast locations were a result of the large inclination gain and occurred near the least effective point on the osculating orbit for reducing Q . Figure 4.7 shows the normalized thrust magnitudes for both cases.

Chapter 5: Low-Thrust Lunar Transfers

This chapter demonstrates how Q-Law can be used for low-thrust Earth-Moon transfers as described in References [58, 59]. The optimal low-thrust Earth to Moon transfer problem has been studied extensively. Many of these efforts employed hybrid methods, where parameter optimization is used to directly reduce the cost function at each iteration and the controls are parameterized by the costate differential equations. Kluever and Pierson used this method to produce spiral trajectories from LEO to LLO in the restricted three-body problem [60], and Ozimek and Howell used the same technique to generate spiral trajectories to periodic orbits in the Earth-Moon system [61]. Direct methods are another common technique for producing spiral trajectories. Herman and Conway used this method to generate low-thrust Earth-Moon transfers [62]. Additionally, Betts used direct collocation to produce time-optimal and mass-optimal solutions for a trajectory problem similar to the SMART-1 trajectory. To solve this SMART-1 test problem, Betts’s initial guess generation efforts required grid scans of velocity-vector thrusting spirals and solving nonlinear optimization subproblems to generate near-feasible guesses [15]. As discussed in prior chapters, another technique for generating low-thrust trajectories is to use Q-Law feedback guidance. Prior Q-Law studies often focused on relatively

simple planetocentric transfers [22, 23, 32, 63, 64], but Jagannatha *et al.* expanded Q-Law’s applicability by using backward propagated Q-Law to design trajectories from GTO to an Earth-Moon halo orbit [65].

This chapter presents a novel application of Q-Law to generate spiral trajectories to the Moon that provides superior initial guesses and enables trade studies that were previously too computationally expensive. This new approach joins forward and backward propagated Q-Law phases at a patch point near the Moon’s sphere of influence to construct end-to-end Earth spiral escape to lunar spiral capture trajectories. This approach is parallelized and wrapped in a Multi-Objective Evolutionary Algorithm to explore the mission design space. The resulting Q-Law solutions provide a strong, feasible initial guess for direct collocation to optimize the trajectory. The Q-Law trajectories are propagated in a full dynamics model, so obtaining a feasible, high-fidelity trajectory for use as an initial guess only requires one Forward-Backward Q-Law evaluation. This represents a significant advancement in Lunar spiral transfer initial guess generation as compared to Betts’s approach [15]. The trajectory optimization uses the same transcription method as in Chapter 4. All trajectory optimization phases are constructed with continuity constraints, resulting in a continuous trajectory within NLP constraint tolerances from the Earth parking orbit to the target Lunar orbit. The dynamics model can have arbitrarily high-fidelity and includes eclipses.

The effectiveness of this new method as an initial guess tool is demonstrated on a problem inspired by the SMART-1 mission, and the results are compared to those of Betts [15]. Then, Forward-Backward Q-Law is demonstrated as a method for per-

forming trade studies by analyzing new mission scenarios with modern departure epochs and an ESPA-class spacecraft. In this first test problem, possible GTO departure epochs and right ascensions were swept through to produce minimum-time trajectories and identify which rideshare scenario makes Lunar capture most difficult. Mass-optimal trajectories were also generated to evaluate possible propellant savings over the minimum-time solutions. In the second test problem, the delivered mass and arrival epoch were constrained. Then using only backward Q-Law, Pareto optimal trajectories were found for a spacecraft departing from Low Earth Orbit (LEO) and arriving at Low Lunar Orbit (LLO). Using these test problems, it is shown that this approach is very enabling for direct optimization efforts and can perform trajectory trade studies that were previously prohibitively computationally expensive. Leveraging this technique allows for efficient trade space exploration and provides feasible, multi-phase initial guesses for optimal trajectory generation.

5.1 Forward-Backward Q-Law

As discussed in Chapter 3, Q-Law cannot effectively target a specific true anomaly, and while Q-Law will generally converge to the desired final orbit, the user will have no control over where on that orbit the spacecraft will arrive. This makes designing Q-Law trajectories to the Moon particularly difficult as a time-varying true anomaly needs to be targeted to achieve proper phasing with the Moon. Additionally, a Lunar orbiter mission will have a target orbit once captured at the Moon, which introduces a second phase to the trajectory design problem. To over-

come the Moon phasing challenge and address the required two trajectory phases, a forward-backward Q-Law approach was introduced.

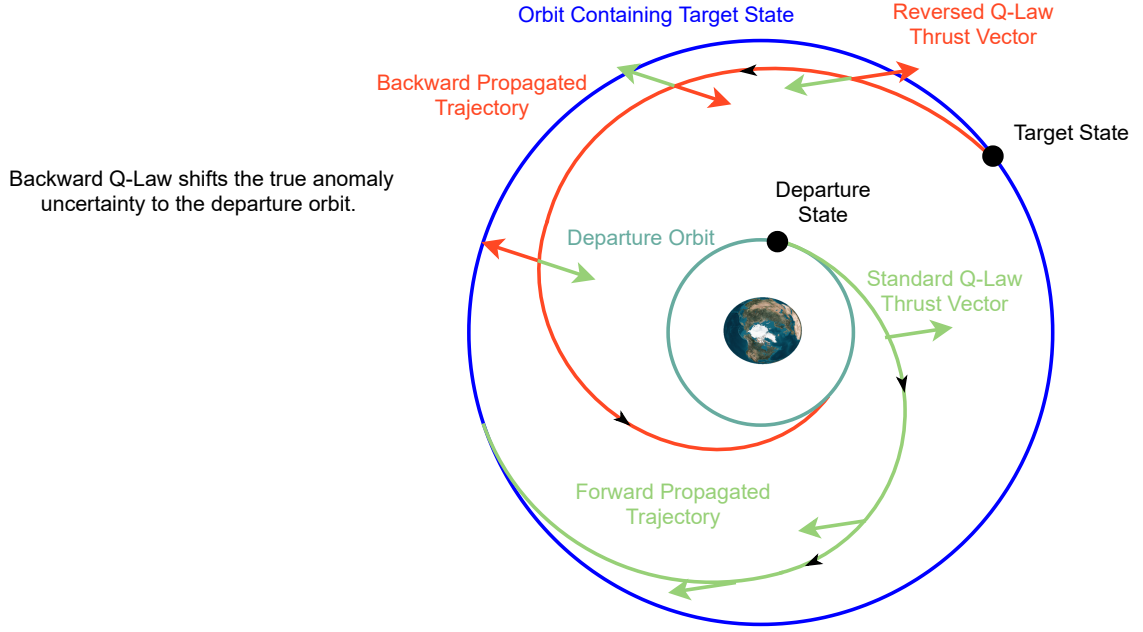


Figure 5.1: Forward and backward Q-Law comparison.

As depicted in Figure 5.1, backward propagated Q-Law moves the true anomaly uncertainty to the departure orbit and can be used to target a specific point on an orbit. By starting at the target state, targeting the departure orbit, and reversing the calculated thrust vector, backward Q-Law will solve for a trajectory that starts somewhere on the departure orbit and ends exactly at the target state.

To solve for spiral trajectories to the Moon, backward and forward propagated Q-Law phases are patched together to target the Moon from the Earth departure orbit and solve for the Lunar spiral. The patch point location is selected near the Moon's sphere of influence to serve as the initial conditions for both the backward and forward Q-Law phases. An illustration of this methodology is shown in Figure

5.2, and the forward-backward Q-Law procedure is summarized below.

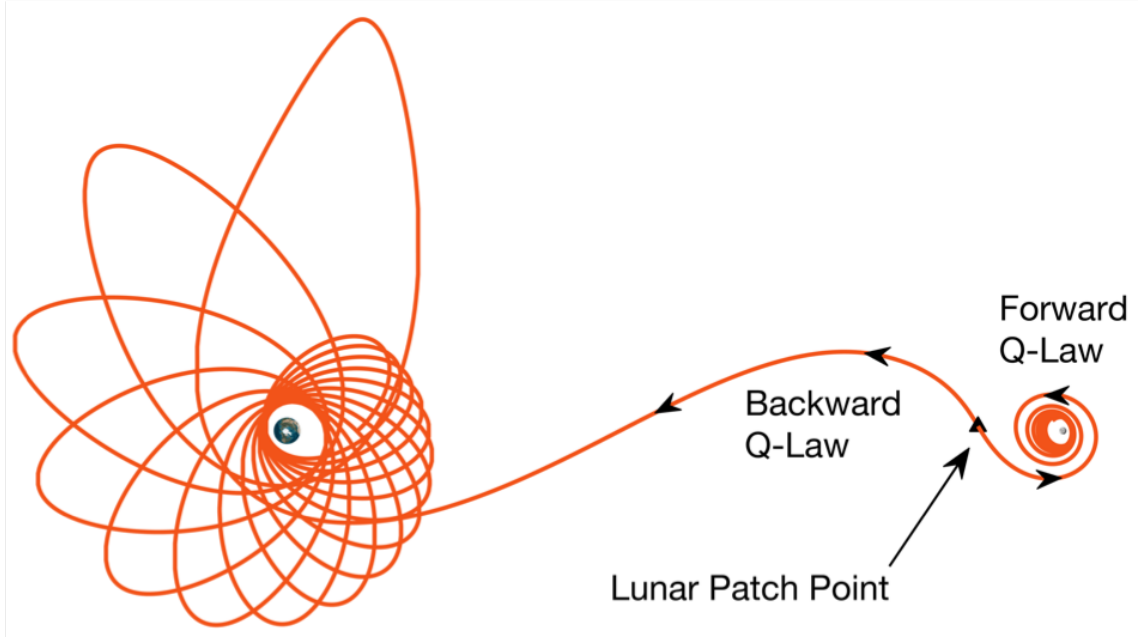


Figure 5.2: Forward-Backward Q-Law depicted in the Earth-Moon rotating frame.

1. Select patch point location, mass, and epoch. Chosen to be loosely captured at the Moon in selenocentric orbital elements.
2. Run forward Q-Law in the selenocentric frame from the patch point to the Lunar target orbit.
3. Convert patch point state to Earth-centered orbital elements.
4. Set Earth-centered path point as initial conditions.
5. Set Earth departure orbit (e.g. GTO) as target.
6. Flip the Q-Law thrust vector within the dynamics model.
7. Propagate backwards in the Earth-centered frame from the target state to the departure orbit.

By leveraging the patch point and backward Q-Law, Q-Law’s inability to target a full state is overcome by shifting the true anomaly variations to the initial and final orbits. All propagation can be executed in an arbitrarily complex dynamics model, resulting in a high-fidelity trajectory from the Earth departure orbit to the Lunar target orbit that is continuous in state, mass, and epoch.

5.2 Optimization Problem Setup

Generating spiral trajectories to the Moon is a two phase optimal control problem, and the Q-Law solutions were used as initial guesses for these phases. The optimal control problem setup follows the one described in Section 4.4 with minor differences in the problem constraints. Equation 5.1 forced the desired initial condition for phase 1 to be the desired Earth departure orbit. The subscript represents which phase the state and time are in.

$$\mathbf{x}_1(0) - \mathbf{x}^{des}(0) = 0 \quad (5.1)$$

Equation 5.2 forced the phase 2 terminal condition to be the desired final orbit at the Moon. The time superscript represents initial or final time, and the state variable superscript indicates the ECI or rotating selenocentric frame. Although an MEE state was used for the optimization, the terminal state was converted back to classical orbital elements to ensure the target orbit is achieved while also allowing for any desired subset of the orbital elements to remain free.

$$\begin{aligned}
a_2^{\mathcal{Q}} \left(t_2^f \right) - a^{des} &= 0 \\
e_2^{\mathcal{Q}} \left(t_2^f \right) - e^{des} &= 0 \\
i_2^{\mathcal{Q}} \left(t_2^f \right) - i^{des} &= 0 \\
\Omega_2^{\mathcal{Q}} \left(t_2^f \right) - \Omega^{des} &= 0 \\
\omega_2^{\mathcal{Q}} \left(t_2^f \right) - \omega^{des} &= 0
\end{aligned} \tag{5.2}$$

Equation 5.3 ensured that the state, mass, and time are continuous through the transition between phases 1 and 2. Here, $^{ECI}\mathbf{Q}^{\mathcal{Q}}$ represents the rotation between the selenocentric and the ECI frames, and $\mathbf{x}_{\mathcal{Q}}^{ECI}$ represents the Moon's state relative to Earth in the ECI frame.

$$\begin{aligned}
\mathbf{x}_1^{ECI} \left(t_1^f \right) - \left[^{ECI}\mathbf{Q}^{\mathcal{Q}} \left(t_2^0 \right) \mathbf{x}_2^{\mathcal{Q}} \left(t_2^0 \right) + \mathbf{x}_{\mathcal{Q}}^{ECI} \left(t_2^0 \right) \right] &= 0 \\
m_1 \left(t_1^f \right) - m_2 \left(t_2^0 \right) &= 0 \\
t_1^f - t_2^0 &= 0
\end{aligned} \tag{5.3}$$

The state and control variable constraints ensured that f, g, h, k stayed within ± 2 and so u_r, u_θ , and u_h stayed within ± 1 . Equations 4.36 and 4.37 or 4.40 and 4.41 were used for time-optimal and mass-optimal problems, respectively. All trajectory

optimization for this problem used IPOPT as the NLP solver. In the mass-optimal problem, IPOPT was found to be superior to SNOPT at producing bang-bang coast arcs on many-revolution trajectories.

5.3 Example: SMART-1 Mission

A trajectory design problem inspired by ESA’s SMART-1 mission was presented and solved by Betts [15]. He successfully produced spiral trajectories from GTO to the Moon using the SMART-1 spacecraft specifications and nominal departure orbit. Betts produced his initial guesses through a velocity-vector thrusting grid scan that searched for the closest approach to the Moon. Then, starting at the final Lunar orbit, an NLP was defined that varied the spacecraft weight, epoch, true anomaly, and right ascension such that when the spacecraft was propagated backwards in time (assuming retrograde velocity-vector thrusting), the discontinuity to the end of the Earth-centered spiral phase was minimized. These two spiral phases were then connected by a Lambert coasting arc. Using this initial guess framework, Betts computed time-optimal and mass-optimal trajectories. Each solution consisted of three phases: a maximum thrust spiral out, a coast arc into Lunar space, and a maximum thrust spiral down at the Moon. Eclipsing was not considered.

To demonstrate the capabilities of the forward-backward Q-Law solution as an initial guess for direct optimization, it was applied to Betts’s trajectory design problem. The problem parameters used to produce the Q-Law solution and setup the optimal control software are shown in Table 5.1. Both mission phases included

Earth, Moon, and Sun gravity, and the Earth spiral out phase included J_{2-4} gravity perturbations, as was used by Betts. The selenocentric frame for the Lunar spiral is defined by the Moon's angular momentum direction, the intersection of the Moon's orbit with the equatorial plane, and the vector that completes the right-hand system. The basis vectors for this frame are defined as

$$\hat{\mathbf{k}}_m = \frac{\mathbf{r}_m \times \mathbf{v}_m}{\|\mathbf{r}_m \times \mathbf{v}_m\|}, \quad \hat{\mathbf{i}}_m = \frac{\hat{\mathbf{k}}_m \times \hat{\mathbf{k}}_e}{\|\hat{\mathbf{k}}_m \times \hat{\mathbf{k}}_e\|}, \quad \hat{\mathbf{j}}_m = \frac{\hat{\mathbf{k}}_m \times \hat{\mathbf{i}}_m}{\|\hat{\mathbf{k}}_m \times \hat{\mathbf{i}}_m\|} \quad (5.4)$$

with $\hat{\mathbf{k}}_e = [0 \ 0 \ 1]^T$ and $\mathbf{r}_m, \mathbf{v}_m$ as the Moon's position and velocity at the reference epoch. The rotation between ECI and selenocentric coordinates is

$$Q = \begin{bmatrix} \hat{\mathbf{i}}_m & \hat{\mathbf{j}}_m & \hat{\mathbf{k}}_m \end{bmatrix}. \quad (5.5)$$

Deviations with a fixed departure epoch and mass can arise because an epoch and spacecraft mass must be selected for the patch point, and after reverse-propagation, there is no guarantee that the terminal mass and epoch will align with the desired departure mass and epoch. For the SMART-1 mission scenario, the Q-Law solution was only used as an initial guess for the optimizer, which easily resolved any errors at the initial guess departure epoch or mass from the desired values. As a result, there was no significant patch point epoch or mass iteration required in the Q-Law design, and the total computation time for the full GTO-Moon Q-Law solution was less than a second. Given the large number of revolutions involved in both the time-optimal and mass-optimal cases, the first optimization iteration was done with

Table 5.1: SMART-1 Problem Parameters

Mission Parameter	
Reference Epoch	Dec 20 2002
m_0	350 kg
Thrust	73.19 mN
I_{sp}	1675.8 s
Dynamics Constants	
g_0	$9.80665 \times 10^{-3} \text{ km/s}^2$
$\mu_{\mathcal{E}}$	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
μ_{\odot}	$1.32712440018 \times 10^{11} \text{ km}^3/\text{s}^2$
$\mu_{\mathcal{L}}$	$4.90486 \times 10^3 \text{ km}^3/\text{s}^2$
$R_{\mathcal{E}}$	6378.14 km
R_{\odot}	695500 km
$R_{\mathcal{L}}$	1737.5 km
J_2	1086.639×10^{-6}
J_3	-2.565×10^{-6}
J_4	-1.608×10^{-6}
Planetary Ephemeris Kernel	de430
Lunar Patch Point Specification (Rotating Selenocentric Frame)	
a	50000 km
e	0.1
i	90.0°
ω	270°
Ω	90°
θ	180°
Mass	275 kg
Epoch	July 31 2003
Backward Q-Law Parameter	
Central Body	Earth
W_a	10
W_e	1
W_i	1
W_ω	1
W_Ω	0
η_a	0
η_r	0
a_{target}	24661.14 km
c_{target}	0.7162279
i_{target}	7°
ω_{target}	178°
α_{tol}	10 km
c_{tol}	0.01
i_{tol}	0.1°
ω_{tol}	0.1°
Forward Q-Law Parameter (ECI Frame)	
Central Body	Moon
W_a	1
W_e	1
W_i	1
W_ω	1
W_Ω	0
η_a	0
η_r	0
a_{target}	7238.0 km
c_{target}	0.6217187
i_{target}	90°
ω_{target}	270°
α_{tol}	10 km
c_{tol}	0.01
i_{tol}	0.1°
ω_{tol}	0.1°
Optimal Control Setting	
NLP Solver	IPOPT
Derivative Type	Central Differences
Collocation Method	RPM-Differentiation
NLP Tolerance	1×10^{-6}
Mesh Method	hp-LiuRao[42]
Mesh Tolerance	1×10^{-4}
Max Collocation Pts	14
Min Collocation Pts	2
l_{scale} (Phase 1)	$1 \times 10^5 \text{ km}$
t_{scale} (Phase 1)	$100 \times 86400 \text{ s}$
l_{scale} (Phase 2)	$1 \times 10^4 \text{ km}$
t_{scale} (Phase 2)	$10 \times 86400 \text{ s}$
L_{scale}	$2\pi \text{ radians}$
m_{scale}	1000 kg
Mesh pts per Revolution (Phase 1)	10
Collocation pts per Mesh (Phase 1)	3
Mesh pts per Revolution (Phase 2)	5
Collocation pts per Mesh (Phase 2)	3

the patch point fixed in space. This allowed the NLP to move the trajectory around the patch point in both space and time as needed without optimizing the transition between the two mission phases, which drastically reduced the complexity of the problem. A solution was found with less than 1 hour of computation time. The second iteration used the prior solution as the initial guess and included a dynamic patch point, resulting in time-optimal and mass-optimal trajectories from GTO to the target Lunar orbit as shown in Figures 5.3 to 5.5. The results are presented in Table 5.2 and compared to Betts’s results.

Table 5.2: Trajectory optimization results for the SMART-1 problem and literature comparison.

	TOF (days)	Final Mass (kg)	Improvement
Betts Time-Optimal	198.38	274.66	-
Time-Optimal	191.97	275.20	6.4 days (3.23%)
Betts Mass-Optimal	201.28	275.00	-
Mass-Optimal	230.0	290.38	15.38 kg (5.59%)

In the time-optimal problem, the solution found using this method outperformed Betts as there is no coast arc present in our solution. For the mass-optimal problem, thrust magnitude was allowed to vary throughout the trajectory, which presents a more challenging optimal control problem than Betts’s approach of varying the Lunar-insertion coast arc duration. The mass-optimal solution demonstrates bang-bang control and involved coast arcs on every revolution, allowing for significant propellant savings over the span of the trajectory.

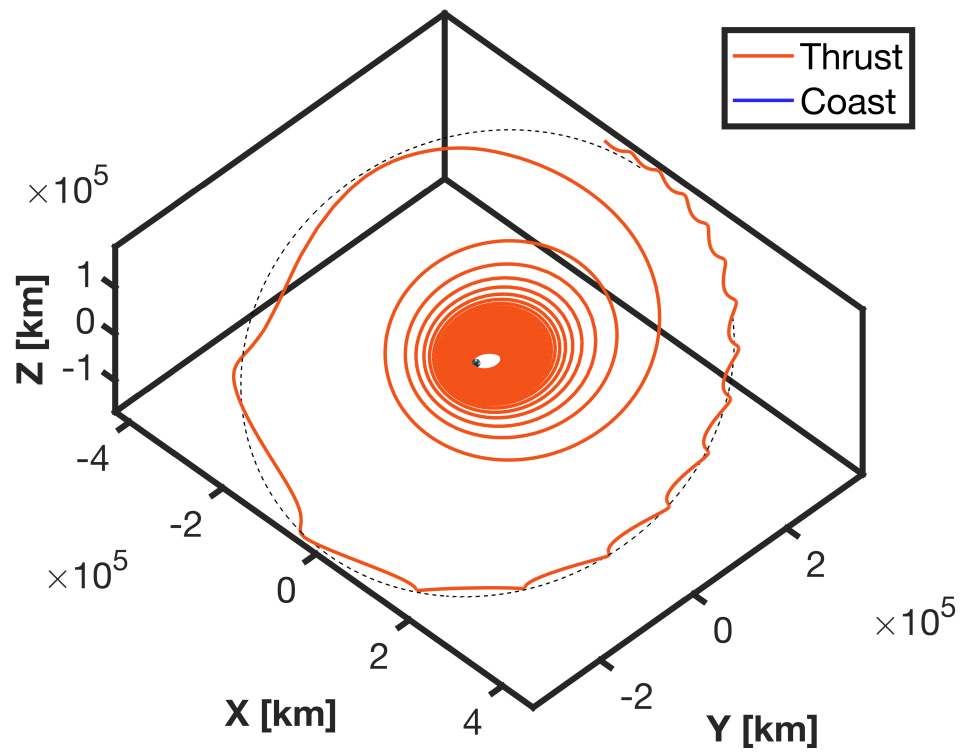


Figure 5.3: Time-optimal SMART-I trajectory in ECI coordinates.

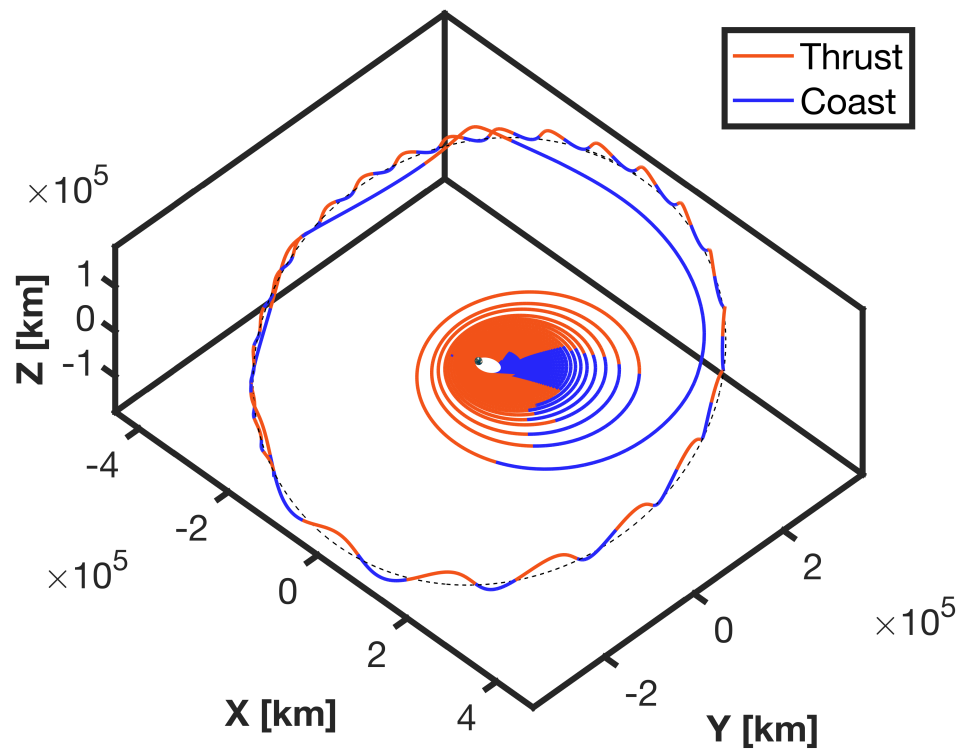


Figure 5.4: Mass-optimal SMART-I trajectory in ECI coordinates.

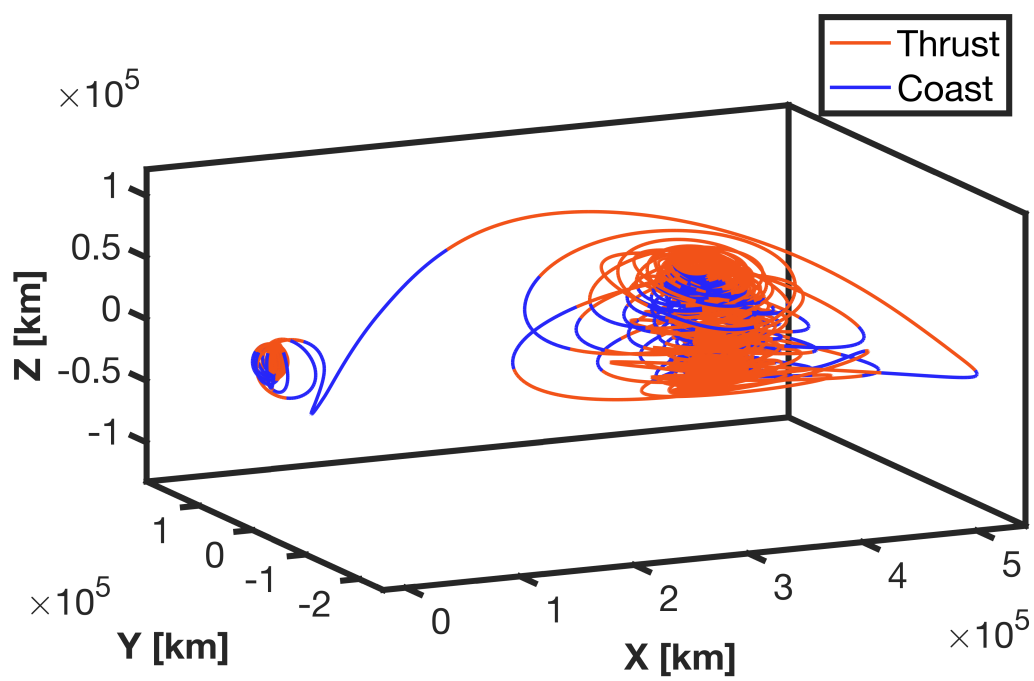


Figure 5.5: Mass-optimal SMART-I trajectory in rotating selenocentric coordinates.

5.4 Example: GTO-Moon Mission

In addition to providing suitable initial guesses for direct optimization, forward-backward Q-Law was found to be a fast and easily parallelizable way to gain valuable insight into the trajectory trade space without extensive optimization. To demonstrate this, a new mission scenario was constructed with a modern departure epoch and ESPA-Class spacecraft with an updated thruster. In this problem, the spacecraft used the Apollo Constellation Engine Max (ACE Max) Hall thruster, which is a 1 kW class Hall Effect thruster currently in development by Apollo Fusion, Inc. [66]. The SMART-1 Lunar target orbit was also used and the Lunar frame was changed to be the body-fixed rotating frame. The new problem and departure orbit parameters are listed in Table 5.3. A thruster duty cycle was included to account for possible thrust errors and constraints that limit thruster levels during the actual mission.

Table 5.3: ESPA-Class Problem specifications.

Mission Parameter	
Year	2025
m_0	180 (kg)
Thrust	60 (mN)
I_{sp}	1760 (s)
Duty Cycle	95 (%)
a_0	24363.99 (km)
e_0	0.7306
i_0	27°
ω_0	0°
Ω_0	free

To simulate a realistic mission design scenario, possible departure right ascensions and epochs were explored to determine the worst case launch scenario. All

propagation was done using the same perturbing forces as the SMART-1 problem with the addition of eclipsing and Lunar J_{2-4} gravity perturbations. When propagating backwards toward Earth, the inclusion of J_2 can make converging to a specific argument of periapsis and right ascension difficult. The spacecraft mass grows with the TOF, and when significant changes in argument of periapsis or right ascension are required, the TOF can increase significantly. To overcome this challenge, a simplifying assumption was made that the spacecraft will follow a velocity-vector thrusting profile until perigee is raised past the radiation belts ($r = 58000$ km), which reduces the size of the trajectory design problem significantly. This technique was used on the original SMART-1 mission [2] and was initially planned for NASA’s DART mission [67]. The backward Q-Law problem was then adjusted such that it matches the osculating orbit achieved after raising perigee. This intermediate orbit was much larger than GTO and was less influenced by aspherical gravity perturbations, thus allowing Q-Law to easily converge. As discussed in Chapter 3, Q-Law cannot target a specific true anomaly. In order to produce a fully continuous trajectory, an additional convergence constraint was added to Q-Law that enforces thrusting until the desired true anomaly on the intermediate orbit is achieved. Note that Q-Law did not adjust the trajectory to meet this true anomaly, rather it thrust to maintain the desired orbit until the target true anomaly is achieved. This method resulted in a trajectory that followed velocity-vector thrusting from GTO to an intermediate orbit just outside the radiation belts, backward Q-Law from that intermediate orbit to the Lunar patch point, and forward Q-Law from the Lunar patch point to the target Lunar orbit, as shown in Figure 5.6.

This trajectory was guaranteed to be continuous in position and velocity but not necessarily in mass and epoch. The discontinuity occurred when the backward Q-Law converged to the terminal state of the velocity-vector spiral because the patch point mass and epoch may not match the target values after reverse propagation. If forward-backward Q-Law is only used for initial guess generation, NLP solvers can easily eliminate these discontinuities. However, in this analysis, it was sought to use Q-Law for trade studies and therefore needed to generate continuous, feasible solutions. To achieve mass and epoch continuity, the forward and backward Q-Law phases were wrapped in the Borg Multi-Objective Evolutionary Algorithm (MOEA) found in the JuliaOpt library [68]. The decision variables were the Lunar patch point location, mass, and epoch and the Q-Law gains for both phases, as listed in Table 5.4. Bounds were placed on the decision vector variables to reduce the search space. The two objectives are shown in Equations 5.6 to 5.7, where Δm and Δt are the mass and epoch discontinuities at the intermediate orbit, respectively. The TOF term represents the sum of the backward and forward Q-Law flight times. $c_1 = 10$ worked well at minimizing errors and yielding a low TOF, and typically the solution with the lowest objective sum was taken as optimal solution. In some cases when the resulting TOF or errors were too large, the MOEA population was evaluated to find a more suitable solution along the Pareto front. In this problem, the solution space was very multi-modal, with many possible combinations of patch point mass and epoch resulting in a small error after backward propagation. Evolutionary algorithms are good at bypassing local minima, and the added TOF objective ensured the solutions

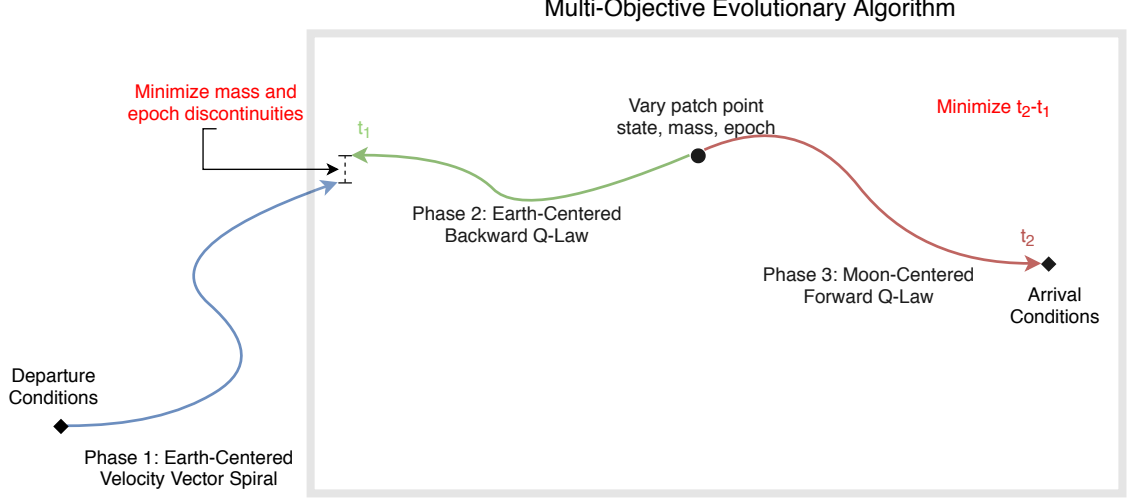


Figure 5.6: Forward-Backward Q-Law optimization procedure.

are feasible and close to time-optimal. All propagation was done using the Tsitouras 5/4 Runge-Kutta method implemented in Julia's differential equation suite [69]. This differential equation suite is very beneficial as the solvers are generally very fast, event functions can easily be incorporated for eclipsing and orbit convergence, and a minimum time-step can be specified for variable-step solvers. This is advantageous because the solver will still meet the required tolerances where it can, and it will not get stuck due to thruster chatter, a common problem when using variable-step solvers for complicated Q-Law trajectories.

$$J_1 = c_1(|\Delta m| + |\Delta t|) \quad (5.6)$$

$$J_2 = TOF \quad (5.7)$$

This process was parallelized on a laptop with 8 GB RAM and a 2.6 GHz

Table 5.4: GTO-Moon Problem MOEA decision vector and bounds.

Patch Point Parameter	Lower Bound	Upper Bound
epoch (days past target epoch)	40	100
mass	120 (kg)	160 (kg)
a	50000.0 (km)	60000.0 (km)
e	0.3	0.8
i	60°	120°
ω	240°	300°
Ω	0°	360°
θ	180°	360°
Earth Phase Parameter	Lower Bound	Upper Bound
W_a	0.1	10.0
W_e	0.1	10.0
W_i	0.1	10.0
W_ω	0.1	10.0
W_Ω	0.1	10.0
Moon Phase Parameter	Lower Bound	Upper Bound
W_a	0.1	10.0
W_e	0.1	10.0
W_i	0.1	10.0
W_ω	0.1	10.0
W_Ω	0.0	0.0

Quad-Core Intel Core i7 processor to sweep through departure right ascensions and epochs. The MOEA was limited to 3000 function evaluations, which typically took 15-20 minutes to produce one complete trajectory with minimal mass and epoch error. It is noted that each Q-Law evaluation took less than 1 second to execute and produce a trajectory to the Moon, and the iteration was only needed to reduce mass and epoch discontinuities. The underlying forward-backward Q-Law method is very computationally efficient and therefore benefits from large-scale parallelization.

The results for the right ascension and departure epoch sweep trajectories are shown in Figure 5.7. As the departure right ascension approaches 180°, the plane change required to properly phase with the Moon increases. These high plane

change scenarios proved to be much more challenging for Q-Law and resulted in longer flight times. The high plane change cases, $130^\circ \leq \Omega_0 \leq 270^\circ$, benefited from a larger MOEA population and additional iterations. A population of 250 with a 5000 function evaluation limit was used to better refine the results in these cases. Cases with departure right ascensions outside this range converged to a low TOF very quickly and exhibited more stable behavior. Figure 5.7a shows bi-weekly oscillations in the minimum TOFs across the calendar year, and Figure 5.7c shows the flight times for $\Omega_0 = 0^\circ$ with the corresponding Lunar distance to Earth at the end of each radiation belt spiral. It is clear that this oscillation is related to the variation in Lunar orbital distance because of its slightly eccentric orbit. Figures 5.7b and 5.7d show the ΔV and required propellant usage for each departure scenario, respectively. Additionally, during the mission planning phase, it may be beneficial to gauge the number of engine cycles needed to fly each trajectory. This parameter sweep provided a useful way to identify possible eclipsing for each departure scenario, as shown in Figure 5.7e. It is noted that for some departure conditions, the spacecraft could achieve capture at the Moon with very little eclipsing.

The Q-Law solutions for the January 1 departure scenario were also used to seed the optimizer to produce time-optimal solutions, as shown in Figure 5.8 for comparison. The associated mass and epoch discontinuities are shown in Figure 5.9. Figure 5.8 shows a departure right ascension around 180° results in the longest TOF for both Q-Law and the optimized solution, representing the worst case departure condition. In all departure scenarios, forward-backward Q-Law produced solutions near the time-optimal solution, with the closest solutions being less than 5% from

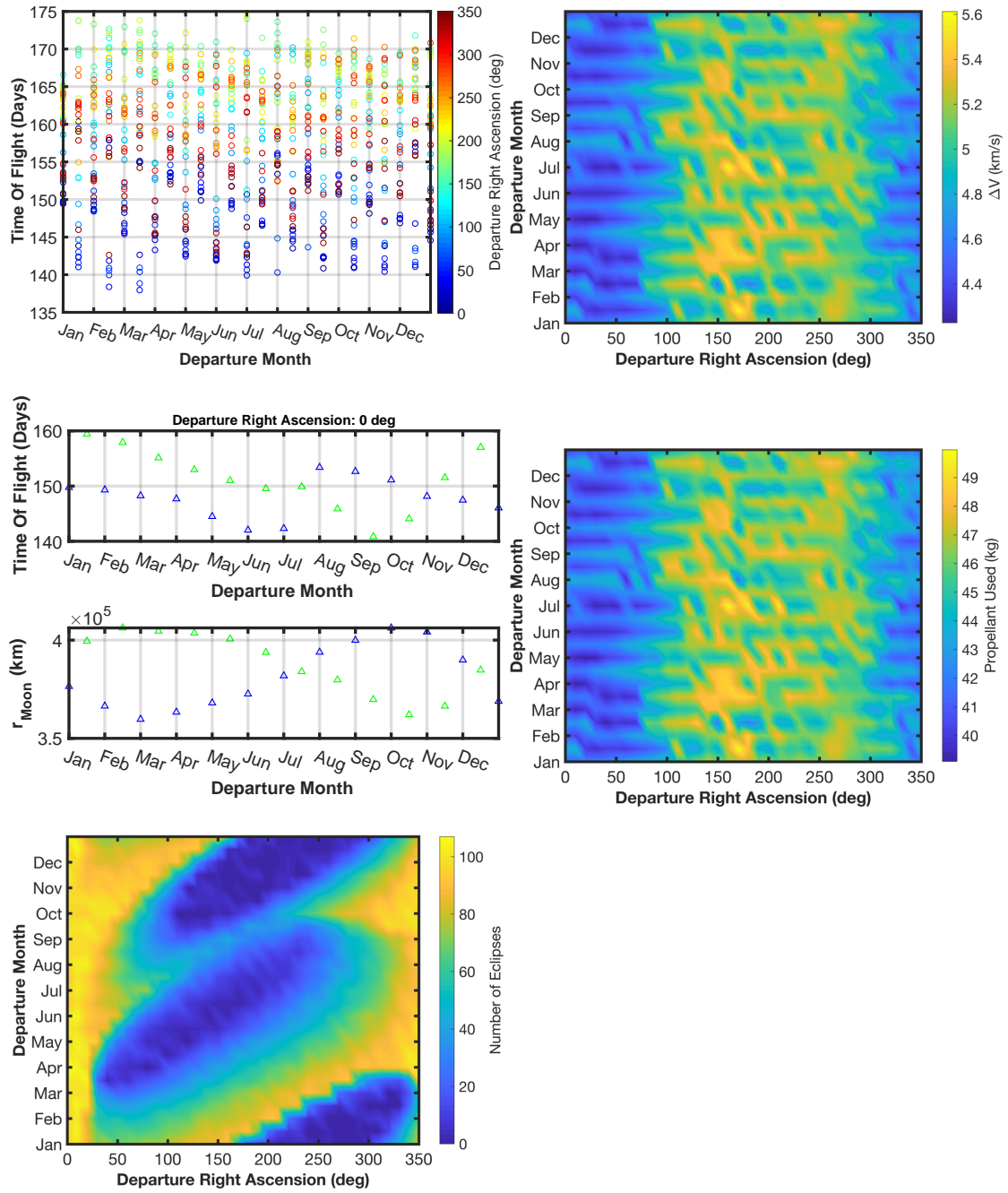


Figure 5.7: Results of the ESPA-Class Mission departure right ascension and epoch sweep.

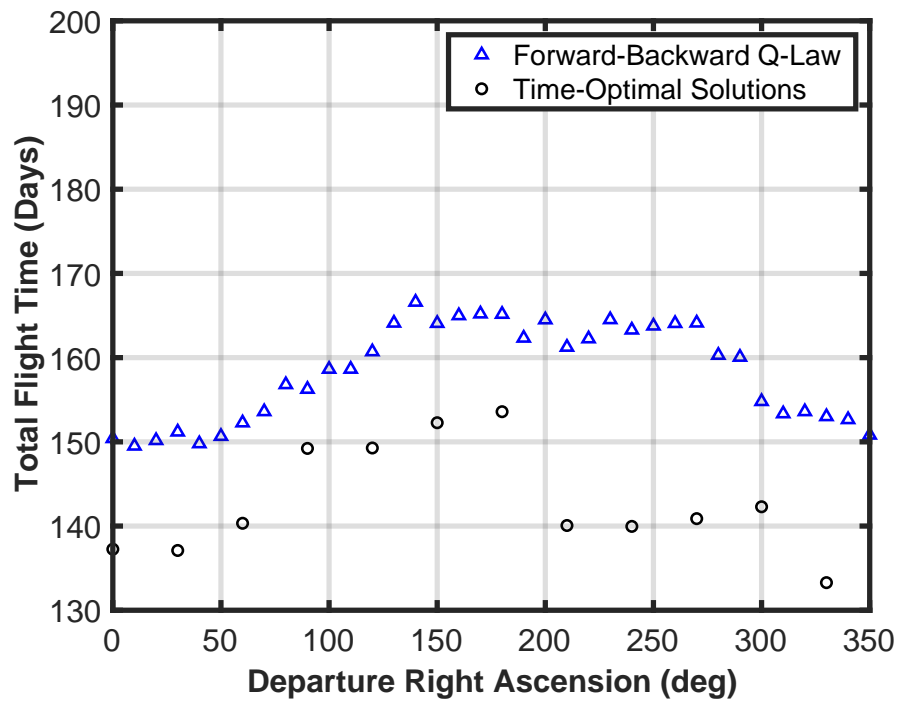


Figure 5.8: Minimum time solutions for the ESPA-Class Mission with a January 1 departure.

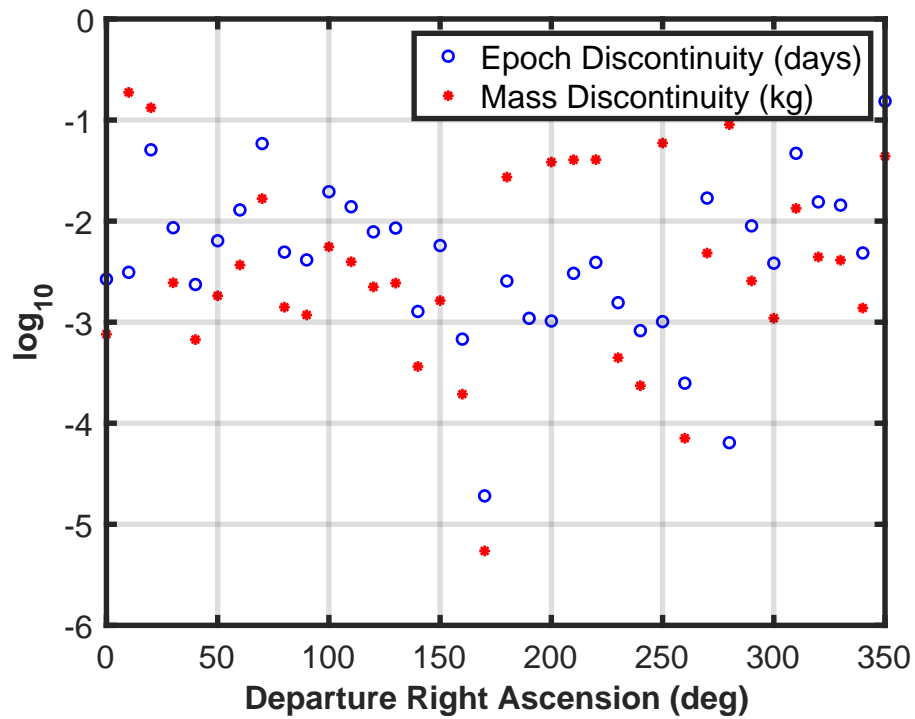
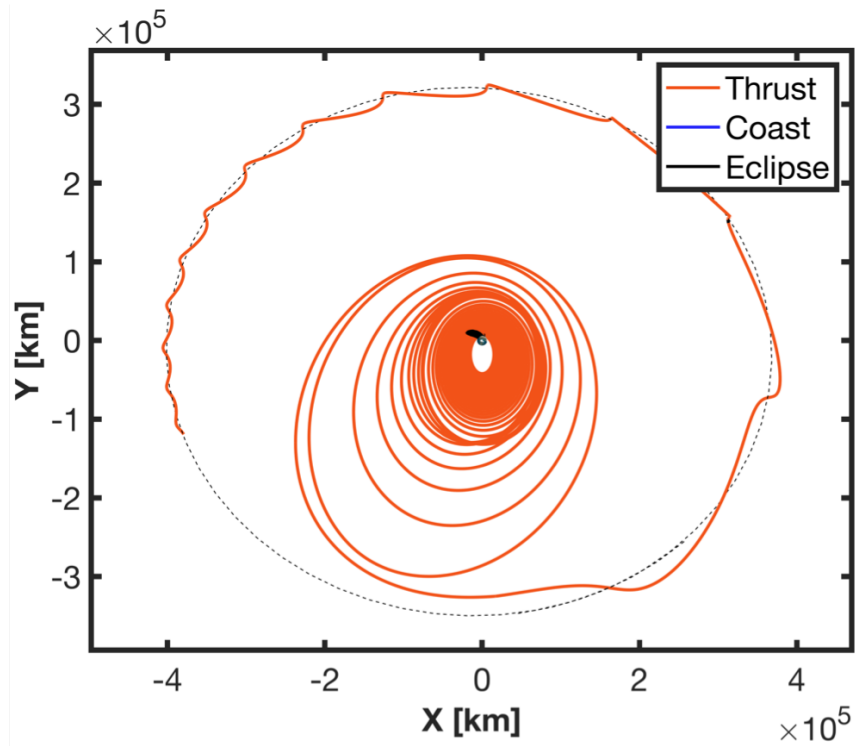
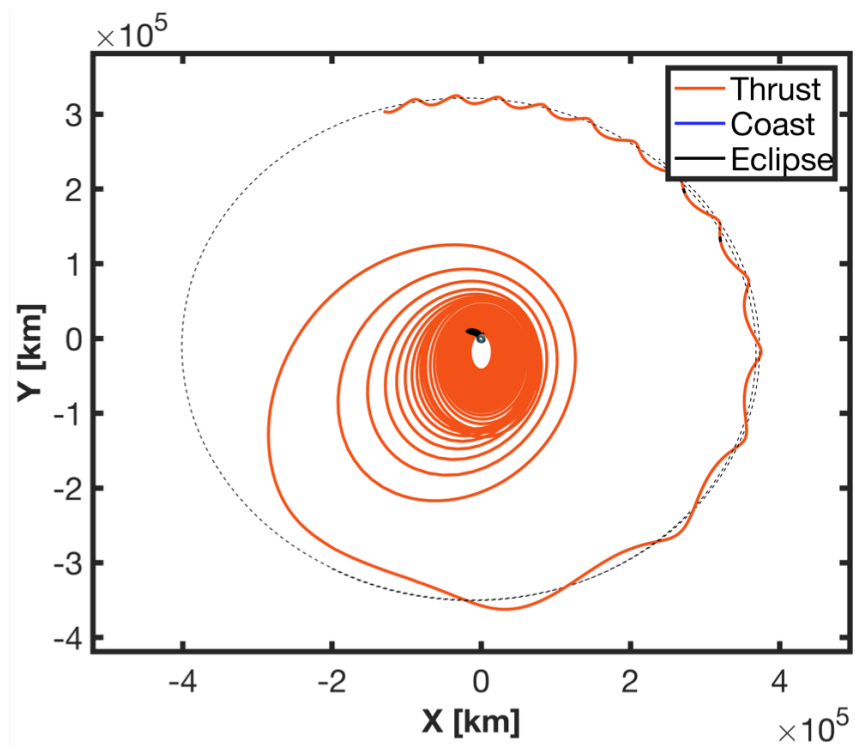


Figure 5.9: Mass and epoch errors for the trajectories shown in Figure 5.8.



(a) Q-Law



(b) Time-Optimal

Figure 5.10: Minimum time solutions for a January 1 departure with right ascension $= 90^\circ$.

the optimal control solution. Q-Law’s TOF is largest with $130^\circ \leq \Omega_0 \leq 270^\circ$ because the required right ascension change is near 180° for these transfers. Unlike Q-Law, the optimal control software was able to optimally leverage 3^{rd} body gravity to reduce the TOF needed to phase with the Moon, which contributed significantly to the resulting TOF differences.

The January 1, $\Omega = 90^\circ$ departure scenario was selected as the test mission scenario to explore possible propellant savings. A comparison of the Q-Law and time-optimal trajectories for this scenario are shown in Figure 5.10. To generate mass-optimal solutions, the same method depicted in Figure 5.6 was used but with Q-Law’s effectivity-triggered coasting included. The effectivity sampling mesh was spaced out every 10 degrees in true anomaly. The absolute effectivity constant was varied on both the backward and forward Q-law phases to produce a TOF-Final Mass Pareto front. The results from the effectivity sweep were also used as the optimizer initial guess to produce time and mass-optimal point solutions from the intermediate orbit outside the radiation belts to the target Lunar orbit.

Q-Law found significant propellant savings when coasting was included, and in some cases, these savings did not require a large TOF penalty, as shown Figure 5.11. When compared to the optimal solutions, the Q-Law results are within a few kilograms of spacecraft delivered mass, and they provided very suitable initial guesses. The trajectory for the mass-optimal point-solution with $\text{TOF} = 163.63$ days, final mass = 146.78 kg is shown in Figures 5.12 and 5.13.

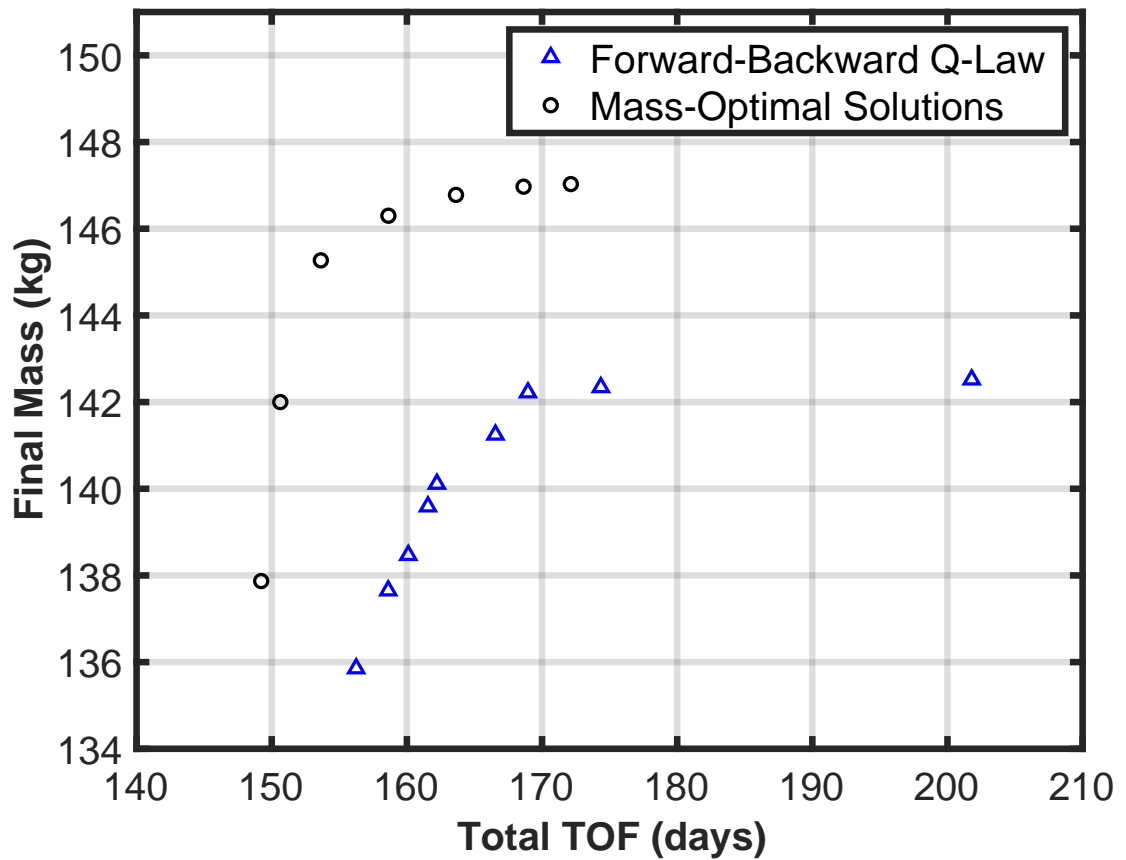


Figure 5.11: ESPA-Class Mission Final Mass vs. TOF.

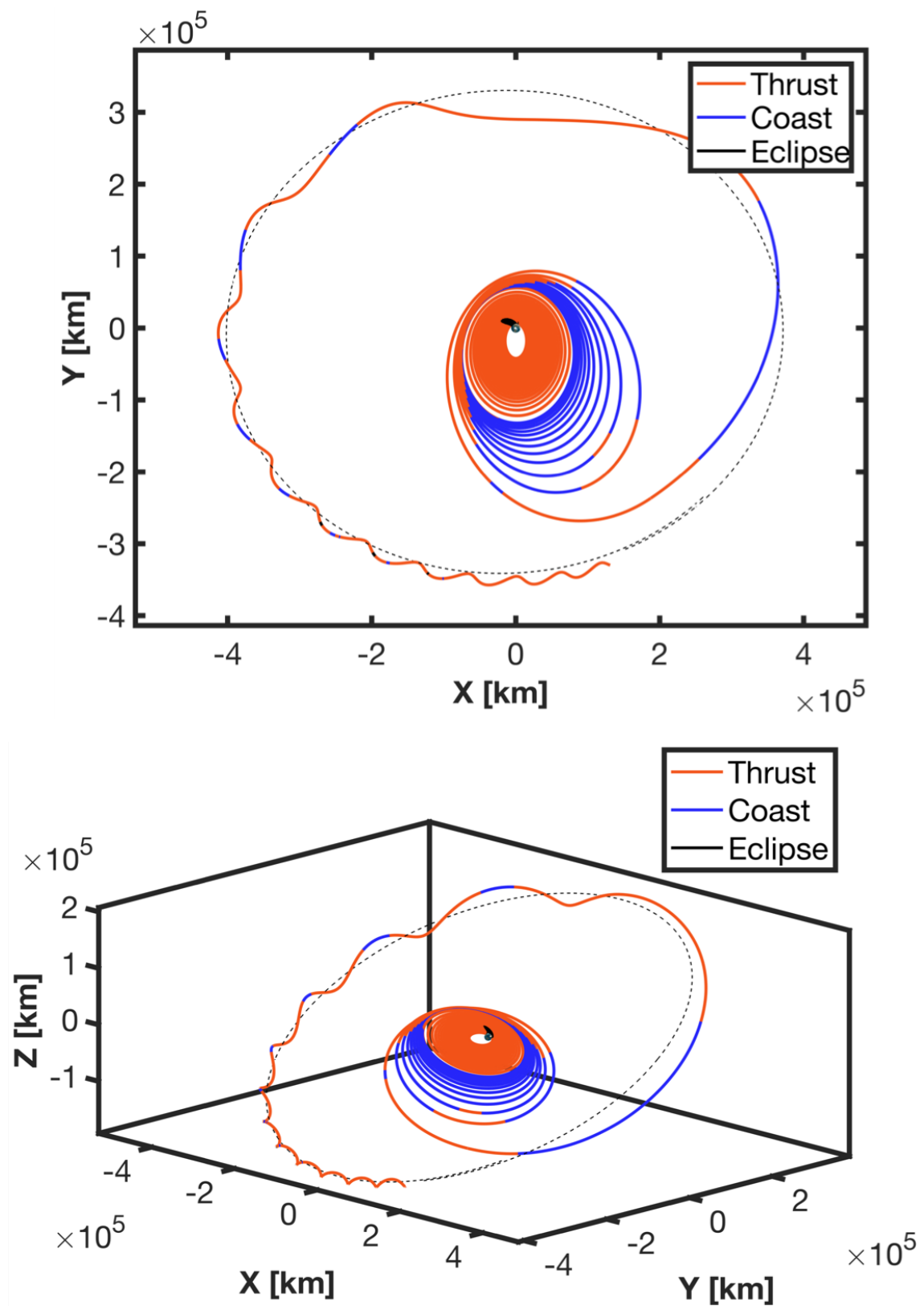


Figure 5.12: Earth-centered mass-optimal trajectory in the inertial frame.

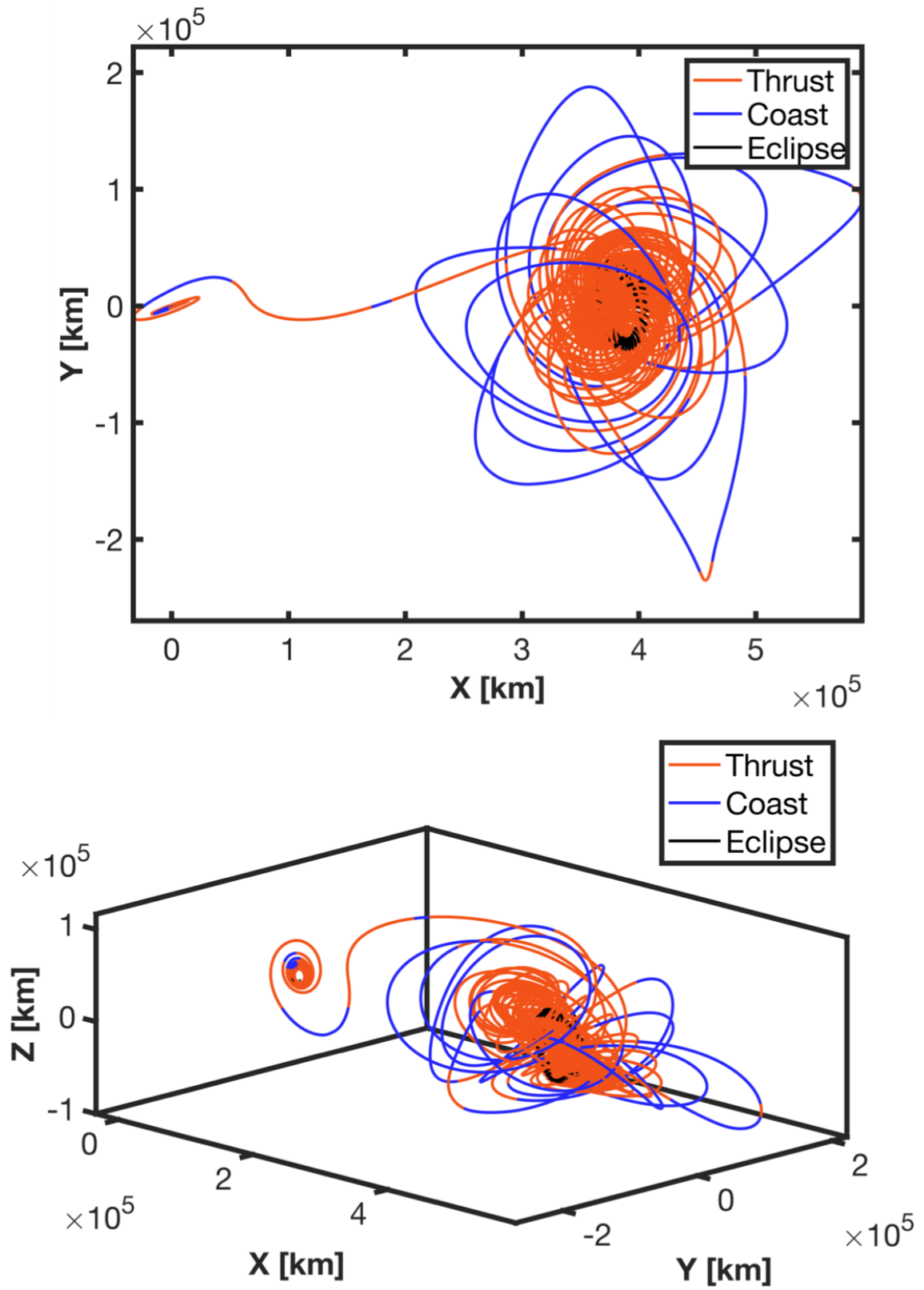


Figure 5.13: Selenocentric mass-optimal trajectory in the rotating frame.

5.5 Example: LEO to LLO Transfer

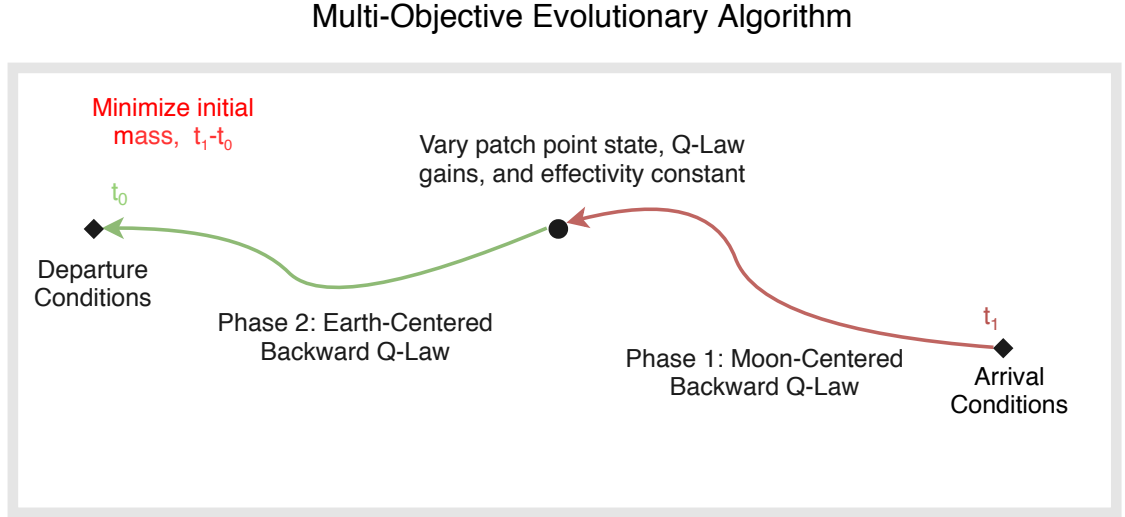


Figure 5.14: All Backward Q-Law optimization procedure.

The next mission scenario explored involved a SmallSat class spacecraft transferring from a circular LEO parking orbit to a polar LLO. In addition to a LEO rideshare, this mission scenario introduces the possibility of leveraging a low-cost, small-lift launch vehicle to deliver the spacecraft into LEO. The small-lift launch vehicle case offers a launch dedicated specifically to the SmallSat and adds significant scheduling flexibility, which may be advantageous for smaller-scale missions. Given the higher ΔV associated with transferring between low-altitude orbits, this spacecraft employed two ACE Max thrusters to reduce flight time. In this problem, the delivered mass and arrival epoch were constrained, representing the scenario where mission/science objectives dictate the required delivered dry mass and arrival epoch at the target Lunar orbit. The LEO departure orbit was constrained in semi-major axis, eccentricity, and inclination, so this scenario assumed the launch

mass can be tailored and phasing orbits in LEO can be used to meet the modeled departure state. To solve this problem, Moon-centered backward Q-Law was used to spiral out from LLO to a patch point, where Earth centered backward Q-Law then took over to complete the spiral to LEO. In the prior GTO-Moon mission scenario, the MOEA was needed to reduce the mass and epoch discontinuities that arise from a fixed departure mass and epoch. An all-backward Q-Law trajectory has no discontinuities as the departure epoch and mass are free to vary. As a result, the MOEA wrapped Q-Law solver can directly trade TOF with propellant usage. To produce the Pareto front, the MOEA varied the Moon-centered Q-Law gains, effectivity parameter, and target orbital elements. It also varied the Earth-centered Q-Law gains and effectivity parameter. This process is shown in Figure 5.14. The decision vector variables and bounds are shown in Table 5.5, and the static problem parameters are shown in Table 5.6. Note that only semi-major axis and eccentricity were targeted in the Moon-centered phase. Omitting the other elements greatly reduced the problem size and produced favorable results compared to when plane change was included. Each function evaluation within the MOEA resulted in a trajectory that starts at LLO and propagates backward to meet the target semi-major axis and eccentricity described by the patch point in the current decision vector. After achieving the target orbital elements, the Earth-centered phase began, which propagated the trajectory from the patch point down to LEO. The MOEA was limited to 10000 function evaluations and used a population size of 200.

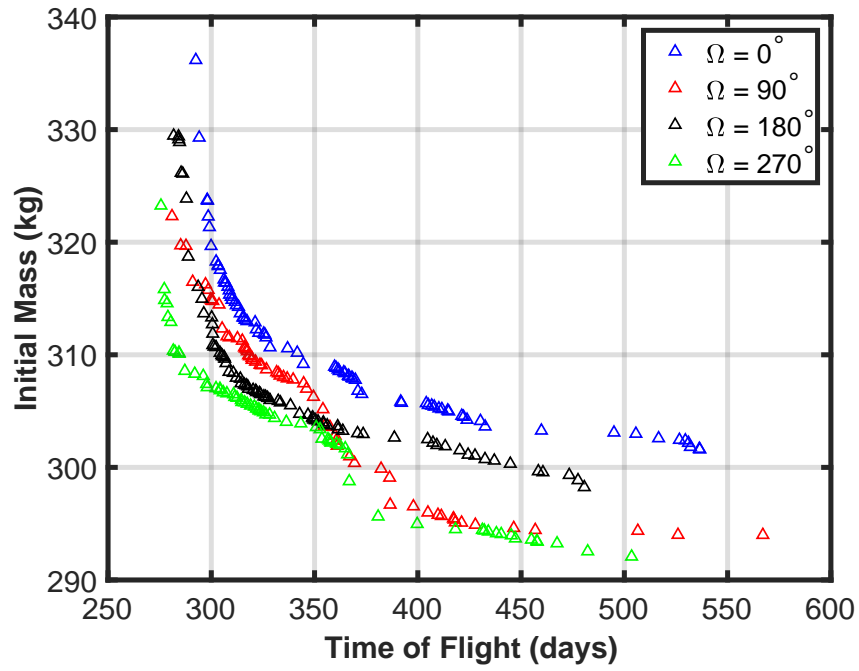
Figure 5.15 shows that differing arrival right ascensions can result in very different propellant requirements, with the $\Omega = 270^\circ$ case performing the best. For

Table 5.5: LEO-LLO Problem MOEA decision vector and bounds.

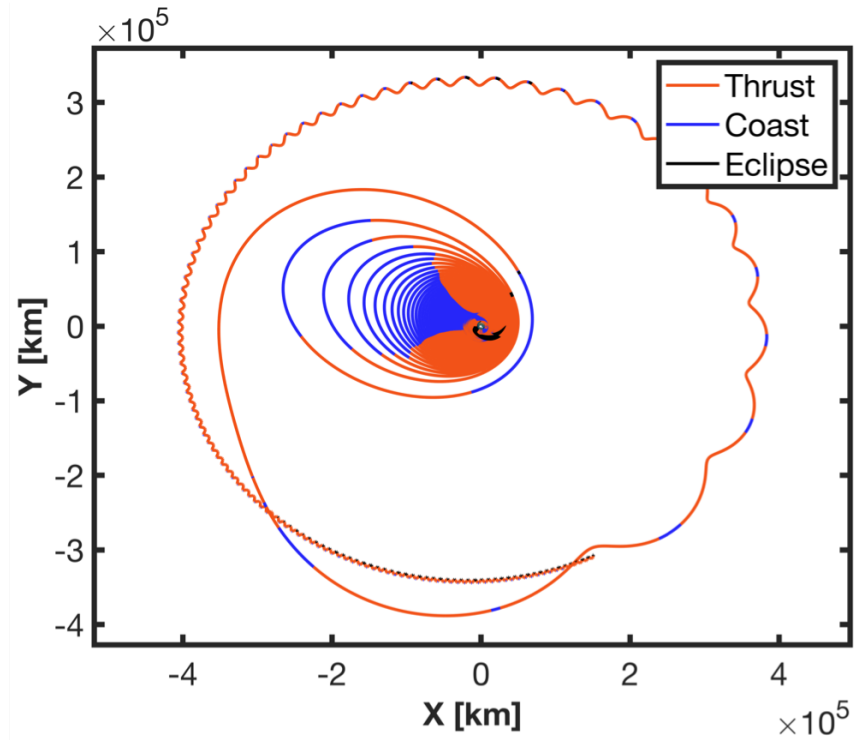
Moon Phase Parameter	Lower Bound	Upper Bound
W_a	0.1	10.0
W_e	0.1	10.0
W_i	0.0	0.0
W_ω	0.0	0.0
W_Ω	0.0	0.0
η_r	0.0	0.5
a_{target}	40000.0 (km)	60000.0 (km)
e_{target}	0.1	0.8
Earth Phase Parameter	Lower Bound	Upper Bound
W_a	0.1	10.0
W_e	0.1	10.0
W_i	0.1	10.0
W_ω	0.0	0.0
W_Ω	0.0	0.0
η_r	0.0	0.5

Table 5.6: LEO-LLO Problem specifications.

Mission Parameter	
Arrival Epoch	Jan 1 2025
m_f	200 (kg)
Thrust	120 (mN)
I_{sp}	1760 (s)
Duty Cycle	95 (%)
LLO Parameter	
a	2000 (km)
e	0.1
i	90°
ω	0°
Ω	free
LEO Parameter	
a	7000.0 (km)
e	0.01
i	27°
ω	free
Ω	free



(a) LEO-LLO propellant usage trades for different LLO right ascensions.



(b) LEO-LLO trajectory for $\Omega = 0^\circ$. TOF = 306 days.

Figure 5.15: Backward Q-Law solutions for the LEO-LLO scenario.

both the $\Omega = 90^\circ$ and 270° arrival orbits, the propellant requirements start to flatten out near $\text{TOF} = 350$ day before quickly dropping off as flight time increases. In all cases, launch masses less than 310 kg are observed, which falls within typical rideshare requirements and the lift capabilities of existing small-lift launch vehicles [70, 71]. This mission scenario presents a challenging trajectory design problem given the large number of revolutions, coast arcs, and eclipses present. This high-fidelity mass trade would have required immense computational effort if it were conducted using other existing optimization techniques like collocation. However, despite the transfer complexity, combining two backward propagated Q-Law phases and a Multi-Objective Evolutionary Algorithm allowed for efficient and direct generation of the propellant usage Pareto front.

Chapter 6: Low-Thrust Lunar Swingby Escape Trajectories

This chapter is based on the work presented in Reference [72]. Gravity assists are well-known to provide significant spacecraft ΔV , and are commonly used in interplanetary missions to alter the spacecraft energy and/or orbit plane [73, 74, 75]. Gravity assist sequences can also be applied to more efficiently escape the Earth-Moon system. This was demonstrated by both STEREO spacecraft, which leveraged Lunar swingbys to escape from highly eccentric geocentric orbits [76]. The double Lunar gravity assist sequence has been shown to be very effective at achieving higher escape C_3 . Yárnoz *et al.* extended the database approach of Lantoine and McElrath to identify many families of double Lunar flyby trajectories [77, 78, 79]. In their work, a database was constructed by searching through the system configuration space in the Sun-Earth Circular Restricted Three Body Problem for feasible, ballistic Moon-Moon transfers. For a given problem, the database was searched for an existing solution that meets the required escape condition, which was then used as the initial guess for a high-fidelity differential corrector.

In this chapter, the focus is on rideshares to GTO where the spacecraft then escapes under its own power by leveraging Lunar gravity assists. Spiral trajectories were generated starting from GTO and ending with the first Lunar gravity assist in

the sequence. This gravity assist either propelled the spacecraft out into interplanetary space or altered its Earth-centered orbit such that it encountered the Moon again for a higher energy escape. These trajectory types were compared with conventional spiral escape solutions for a flyby mission of Comet 45P. All spirals were generated from backward propagated Q-Law [22, 23, 59], and the Lunar swingby and interplanetary phases were designed using the Evolutionary Mission Trajectory Generator (EMTG) [80].

6.1 Perturbed Sims-Flanagan Transcription

The low-thrust model used for the non-spiral portion of this study was based on the Sims-Flanagan transcription and is shown in Figure 6.1 [81]. The Sims-Flanagan transcription discretizes the trajectory, with the boundaries of each phase serving as a control point. A phase has two parts to it. The first half of the trajectory begins with forward propagation from the previous control point, and the second half begins with backward propagation starting at the next control point. Continuity constraints are enforced at the match point between the two parts. Phases are discretized into N equally-sized time steps, and the spacecraft state is propagated between time steps by solving Kepler’s equation. Within each time step, impulsive velocity changes are applied to the spacecraft’s state as an approximation to the continuous thrusting of a low-thrust engine. The ΔV impulses are limited such that they cannot exceed the ΔV achievable by the continuously thrusting engine over the course of one time step. Gravitational perturbations from additional bodies are

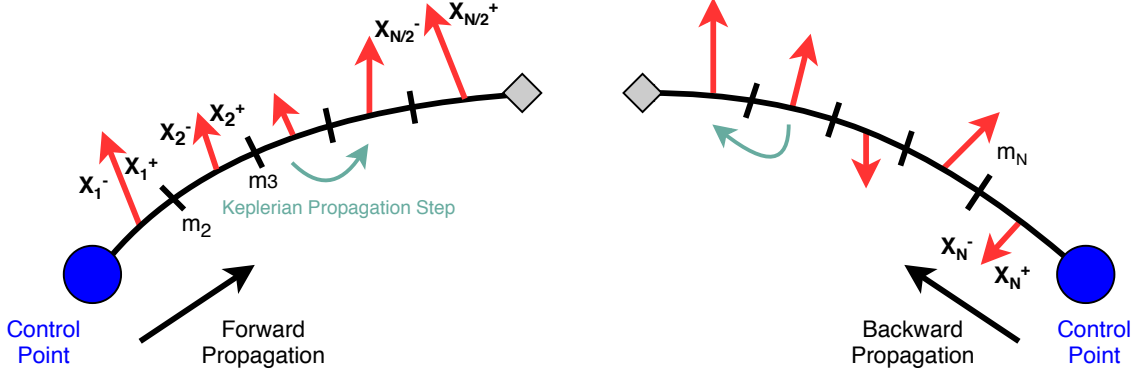


Figure 6.1: A single Sims-Flanagan phase divided into N time steps.

also included in the ΔV applied to the spacecraft to better represent the dynamics [82, 83].

The forward and backward propagation ΔV impulses were calculated using Equation 6.1 and Equation 6.2, respectively.

$$\mathbf{v}_k^+ = \mathbf{v}_k^- + \frac{N_{active}DT_{max}\Delta t_k}{m_k}\mathbf{u}_k + \sum_{p=1}^n -\frac{\mu_p}{r_p^3}\mathbf{r}_p\Delta t_k \quad (6.1)$$

$$\mathbf{v}_k^- = \mathbf{v}_k^+ - \frac{N_{active}DT_{max}\Delta t_k}{m_k}\mathbf{u}_k - \sum_{p=1}^n -\frac{\mu_p}{r_p^3}\mathbf{r}_p\Delta t_k \quad (6.2)$$

Here, N_{active} is the number of active thrusters, D is the thruster duty cycle, T_{max} is the maximum available thrust for the current time step, $\Delta t_k = \frac{t_f - t_0}{N}$, μ_p is the gravitational parameter of the perturbing body, r_p is the distance between the spacecraft and that body at the time of the impulse, and \mathbf{u}_k is the k^{th} time step control vector. The spacecraft mass at the k^{th} time step was calculated as

$$m_k = m_{k-1} \pm \|\mathbf{u}_{k-1}\| D\Delta t_k \dot{m}_{max_{k-1}} \quad (6.3)$$

with a + sign taken during the backward propagation. The mass flow rate was calculated using Equation 2.10, and the optimization problem was constrained to ensure more realistic control and dynamics. The control vector at each time step was constrained such that

$$\|\mathbf{u}_k\| = \sqrt{u_{x_k}^2 + u_{y_k}^2 + u_{z_k}^2} \leq 1 \quad (6.4)$$

Additionally, match point constraints were enforced to ensure continuity between the forward and backward propagated sections of the trajectory. The match point constraints used the terminal states after forward and backward propagation and were formulated as:

$$\mathbf{c} = \mathbf{X}_{mp}^F - \mathbf{X}_{mp}^B = \begin{bmatrix} \mathbf{r}^B - \mathbf{r}^F \\ \mathbf{v}^B - \mathbf{v}^F \\ m^B - m^F \end{bmatrix} = \mathbf{0} \quad (6.5)$$

Gravity assists, like the one diagrammed in Figure 6.2, were modeled by enforcing the patched conic approximation shown in Equations 6.6 to 6.8.

$$v_{\infty+} - v_{\infty-} = 0 \quad (6.6)$$

$$\delta - \pi - 2 \arccos \frac{1}{1 + \frac{r_{min} v_{\infty}^2}{\mu}} \leq 0 \quad (6.7)$$

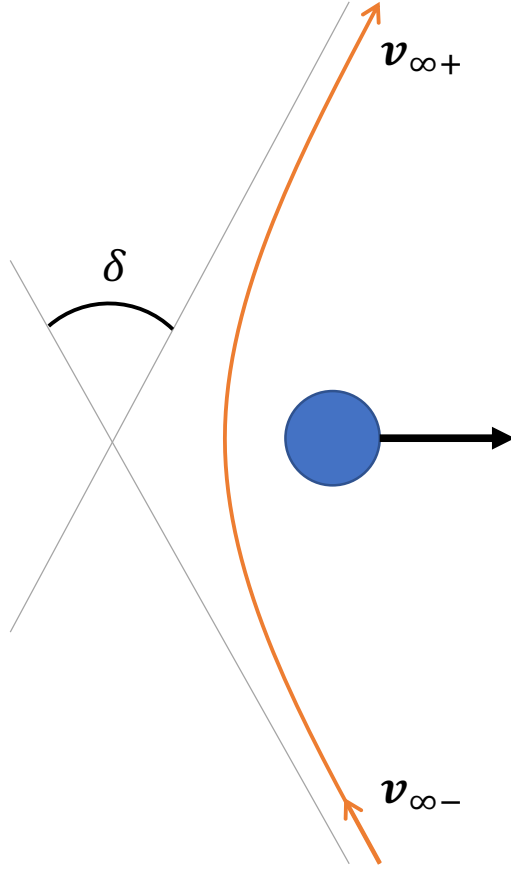


Figure 6.2: Planetary gravity assist.

$$\delta = \arccos \frac{\mathbf{v}_{\infty-} \cdot \mathbf{v}_{\infty+}}{v_{\infty-} v_{\infty+}} \quad (6.8)$$

6.2 Monotonic Basin Hopping

Monotonic Basin Hopping (MBH) is a global optimization heuristic method that explores the entire design space. MBH identifies locally optimal solutions and searches for improvements in other nearby solutions through stochastic variation. The MBH process consists of two loops: an inner loop and an outer loop. The inner loop evaluates the current design variables and consists of a gradient-based opti-

mization scheme that identifies a locally optimal solution. NLP solvers have been successfully used for this step and have proven to be very effective when applied to spacecraft trajectory optimization [84, 85]. The outer loop selects values for the design variables by creating random perturbations on the current best decision vector, i.e., the decision variables that have produced the best NLP cost function thus far. By perturbing the current best decision vector, MBH moves the solution guess away from the current local optima in search of a better solution basin, allowing the NLP solver to identify a more optimal solution. This process is depicted in Figure 6.3.

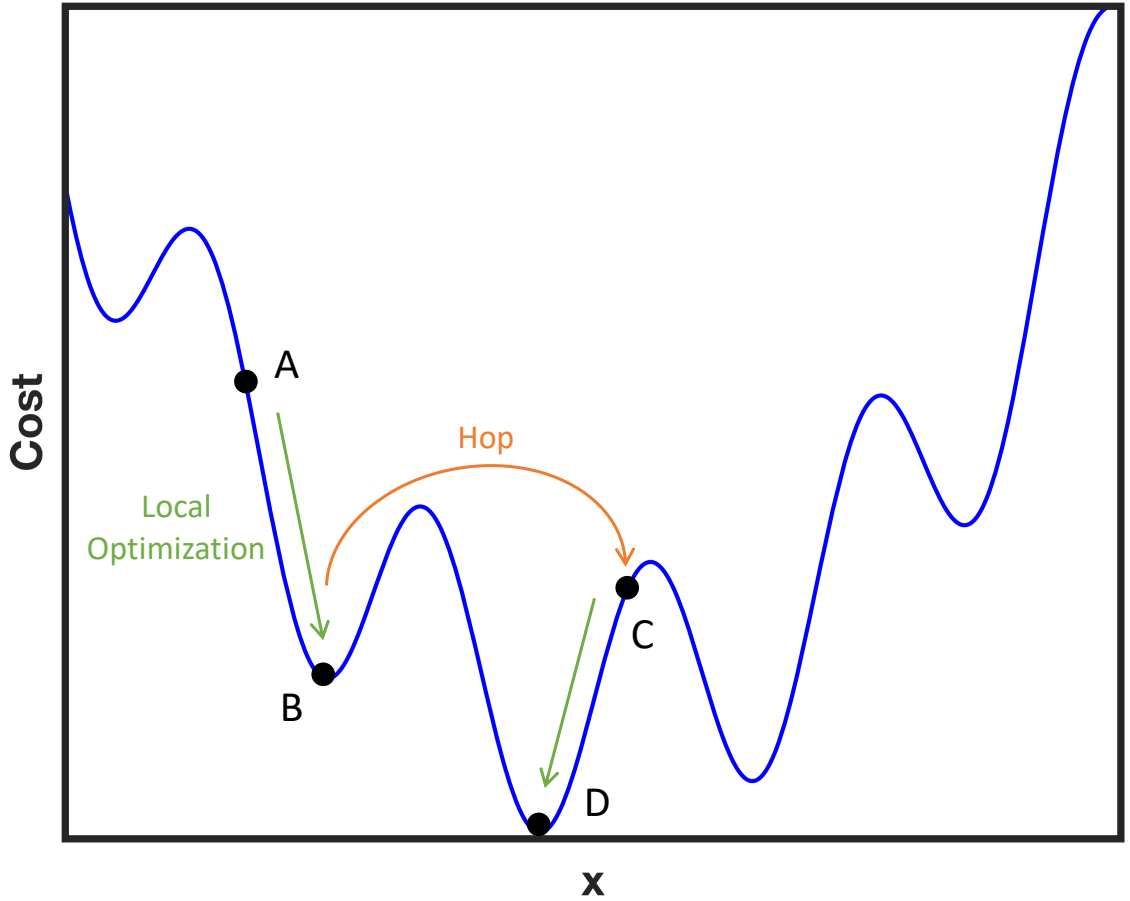


Figure 6.3: Monotonic Basin Hopping procedure.

6.3 Lunar Swingby Design

Prior work on the double Lunar swingby problem has shown that Solar gravity perturbations are key to achieving a naturally occurring V-infinity Leveraging Maneuver (VILM) [77, 86]. After the first swingby changes apoapsis, the solar perturbations can have significant effects on the spacecraft at large orbital distances. This ΔV can alter the encounter v_∞ of the second gravity assist, making it more efficient for achieving escape. As a result, including these perturbations in the Sims-Flanagan model is important for generating the Moon-Moon legs in the double swingby sequence.

In this work, both single and double Lunar swingbys were explored. In the single flyby case, the initial state was bounded at the Moon’s position, and a maximum impulse bound was applied to limit the outgoing v_∞ . An Earth centered coast phase takes the spacecraft to the Earth-Moon Sphere of Influence (SOI), with the remaining mission phases computed using Sun-centered Sims-Flanagan. MBH was applied to the trajectory optimization decision vector to search the design space. The perturbed Sims-Flanagan model paired with MBH generally identified single flyby solutions very quickly.

The double swingby problem is much more difficult than the single flyby case to solve, and generating an initial solution can be challenging as the Moon-Moon legs are complicated to design and many families of transfers exist [77]. However, it was found that leveraging high v_∞ single flyby solutions can be a beneficial first step to design the double flyby sequence. In this approach, a high v_∞ single flyby trajectory

was designed first. Next, the Moon-Moon leg was constructed. In the Moon-Moon leg, the initial position and v_∞ were constrained as before, but now the subsequent Earth-centered Sims-Flanagan phase targeted the second Lunar encounter for a Zero Sphere of Influence (ZSOI) flyby that matched the conditions identified by solving the high v_∞ single flyby problem. This approach reduced the search space by constraining the second Lunar encounter, allowing the MBH to easily identify an initial solution that ensures the flyby sequence results in an advantageous Earth escape condition. After the second flyby, the spacecraft exited the Earth’s SOI into interplanetary space, similar to the single flyby case. Thrusting was allowed in the Moon-Moon leg so the spacecraft can perform its own low-thrust VILMs in addition to the perturbations experienced from the Sun. This was advantageous as the spacecraft did not need to rely on Solar perturbations as the only source of the ΔV needed before the next Lunar encounter. This can reduce Moon-Moon flight time and transfer complexity over ballistic solutions.

After using this approach to generate an initial feasible solution, the Perturbed Sims-Flanagan Transcription was used with MBH to evolve the entire trajectory into a more optimal solution. Then, once an optimal low-fidelity solution was identified, it was used as an initial guess for a higher-fidelity two-point shooting model. The two-point shooting transcription is almost identical to the low-fidelity Sims-Flanagan model. The exception is that instead of using Kepler propagation and treating the thrust and natural perturbations as bounded impulses, the trajectory is propagated with an 8th order Runge-Kutta integrator with a realistic acceleration model that includes both the thrust and n-body gravity perturbations.

This approach proved to be very effective at generating double Lunar swingby sequences in EMTG. Figure 6.4 illustrates sample trajectories for each step in the double swingby design process in the Sun-Earth rotating frame. A description of the corresponding steps in the double swingby search process is summarized in the list below.

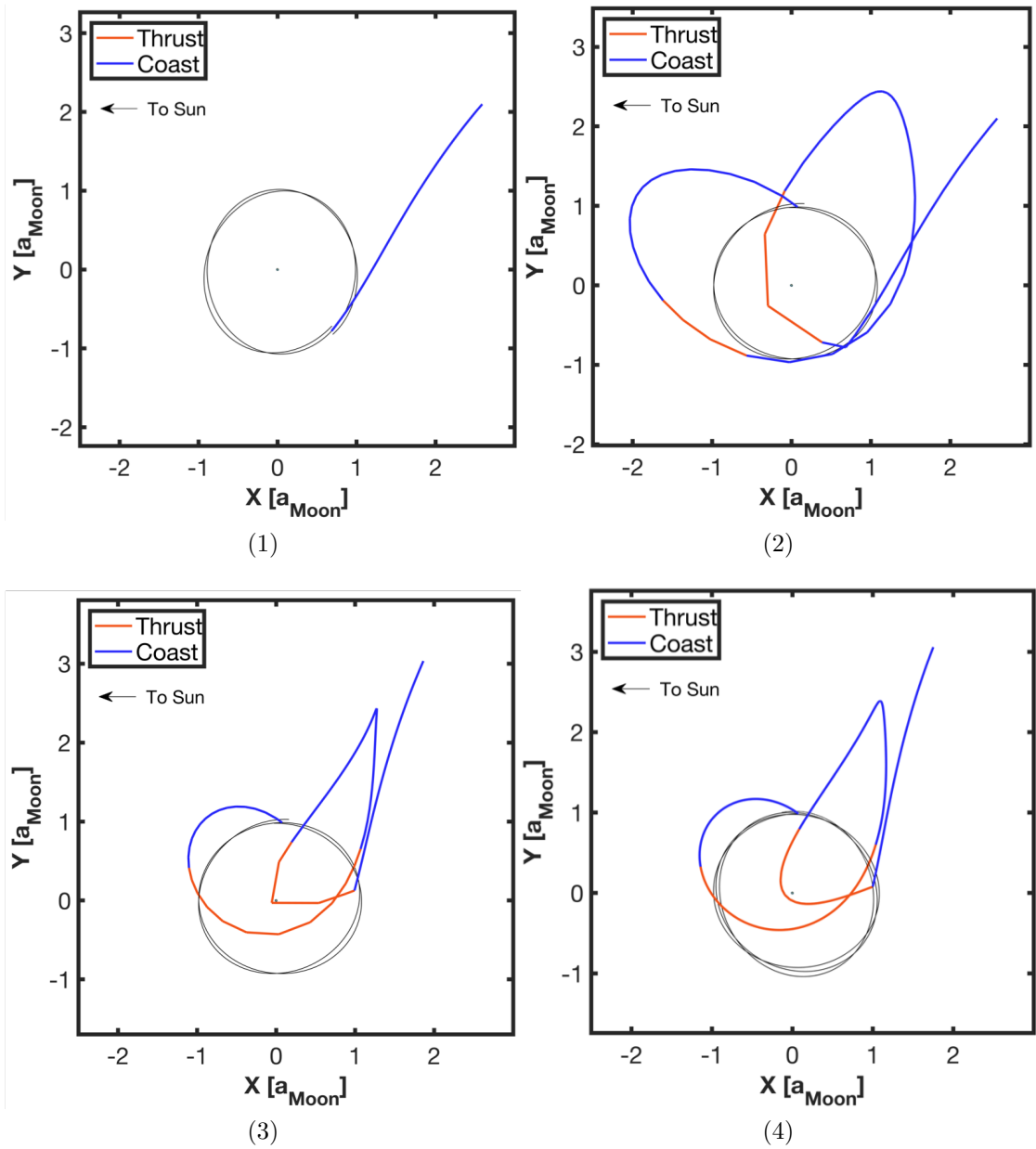


Figure 6.4: Double Swingby trajectory design process.

1. Generate a Lunar departure high v_∞ single flyby trajectory with perturbed Sims-Flanagan and MBH.
2. Use perturbed Sims-Flanagan and MBH to find a low-thrust Moon-Moon transfer that targets the flyby conditions identified in Step 1.
3. Evolve the entire trajectory using perturbed Sims-Flanagan and MBH.
4. Refine the solution further using Runge-Kutta two-point shooting and MBH.

To begin the backwards Q-Law propagation, the pre-flyby state was needed. To determine this, a small NLP was solved to minimize the pre-flyby, Earth-centered Q . The flyby was modeled as ZSOI, and the patched conic approximation described in Equations 6.6 to 6.7 was enforced. The NLP solver chose $\mathbf{v}_{\infty-}$, thus fully defining the pre-flyby state with respect to the Moon. The objective, Q , was evaluated after converting that state to Earth-centered orbital elements. Minimizing the pre-flyby, Earth centered Q subject to the ZSOI flyby constraints in Equations 6.6 and 6.7 ensured the pre-flyby Earth-centered osculating orbital elements were as close to the departure orbit as possible, allowing for a more efficient transfer. This calculation was completed as an intermediate step after EMTG produced a solution.

When propellant usage trades were desired, the Q-Law propagation was wrapped within the Borg Multi-Objective Evolutionary Algorithm (MOEA) implemented in the JuliaOpt library. The MOEA varied the Q-Law gains and effectivity parameters to introduce propellant saving coast arcs. This approach was found to be very effective at generating propellant usage trades.

EMTG is written in C++, so Q-Law was interfaced with EMTG’s output through the Python EMTG Automated Trade Study Application (PEATSA) [87]. PEATSA allows for automated trade space exploration for any mission parameters chosen by the user. The PEATSA studies were parallelized across multiple cores and setup to call Q-Law after EMTG returns a solution for the case at hand. PEATSA evaluated the quality of a run based on the propellant usage across Q-Law’s spiral phase and EMTG’s interplanetary phase.

In this work, spiral escape trajectories were also generated for comparison to the Lunar gravity assist cases. To generate these trajectories, the EMTG problem was modified such that the interplanetary phase initial state was placed at the Earth SOI with a $C_3 = 0$. The Q-Law propagation starts at the SOI state and works backward down to GTO, resulting in a continuous spiral trajectory that starts in GTO and ends at the Earth SOI state identified by EMTG.

6.4 Example: Comet 45P Flyby

These approaches were used to generate flyby trajectories with Comet 45P for a SmallSat spacecraft departing from GTO. Three different escape scenarios were considered: conventional spiral escape with a $C_3 = 0$, single Lunar flyby, and double Lunar flyby. The mission parameters used to solve this problem are listed in Table 6.1. The spiral phase dynamics includes Luni-Solar gravity perturbations, J_{2-4} , and eclipsing.

Figure 6.5 shows some of the double Lunar gravity assist families identified

Table 6.1: Comet 45P Flyby Problem specifications.

Mission Parameter	
Flyby Epoch Bounds	Aug 1 2032 - Jan 31 2033
m_f	100 (kg)
Max Flyby Velocity	6 (km/s)
Thrust	38 (mN)
I_{sp}	1370 (s)
Duty Cycle	90 (%)
GTO Parameter	
a	24363.99 (km)
e	0.7306
i	28.5°
ω	free
Ω	free
Q-Law MOEA Parameter	Bounds
W_a	0.1 - 10.0
W_e	0.1 - 10.0
W_i	0.1 - 10.0
η_r	0.0 - 0.3
η_a	0.0 - 0.3

by the MBH in the Sun-Earth rotating frame. In each case, the outgoing v_∞ for the first gravity assist in the sequence was varied, and each solution included low-thrust VILM's to influence the second Lunar encounter. Figure 6.6a shows the interplanetary phase for the double swingby scenario, and Figure 6.6b shows a point solution for the spiral escape scenario in the ECI frame. Figures 6.6c to 6.6d show single and double swingby solutions in the Sun-Earth rotating frame.

Figure 6.7 shows the TOF-propellant usage trades for the single and double swingby cases as well as the escape spiral scenario. The gravity assist cases were computed with v_{∞_0} varied between 0.4 km/s and 0.7 km/s. As expected, the double swingby case required the least propellant to achieve the flyby. The second swingby enabled a higher Earth escape C_3 making the interplanetary legs more efficient. As v_{∞_0} was increased, the propellant cost for the mission was generally reduced, with

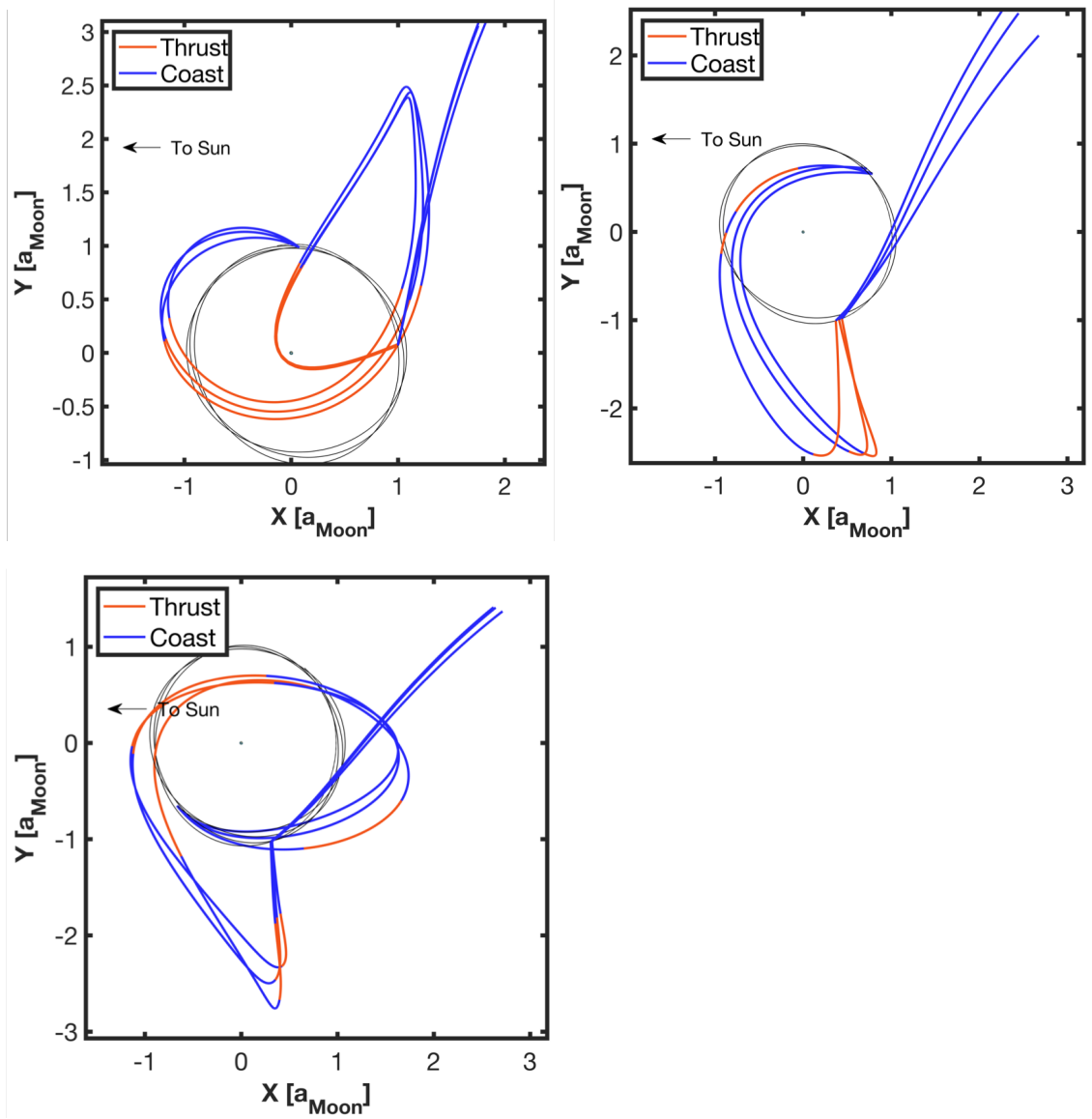
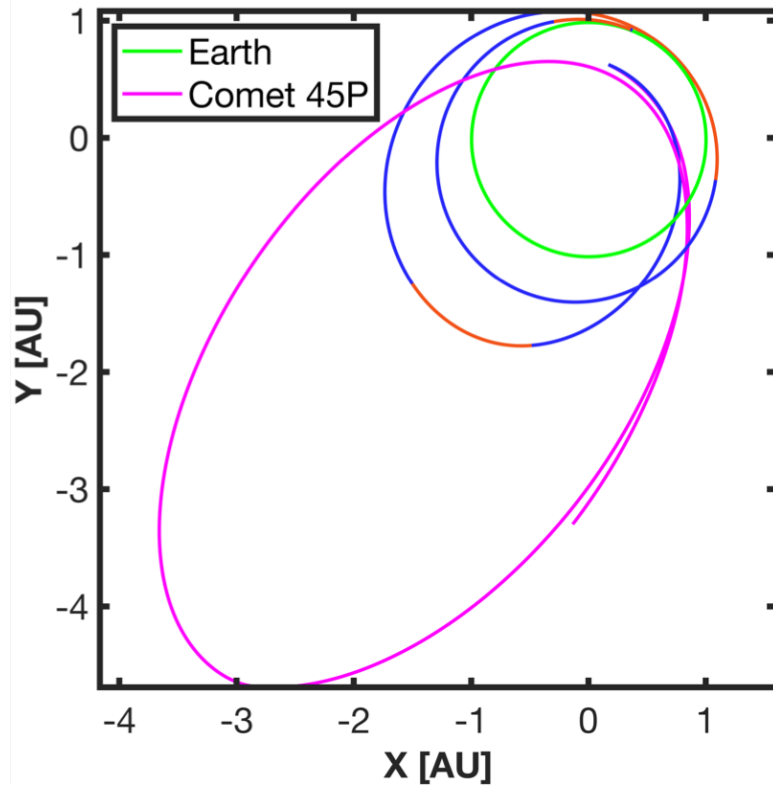
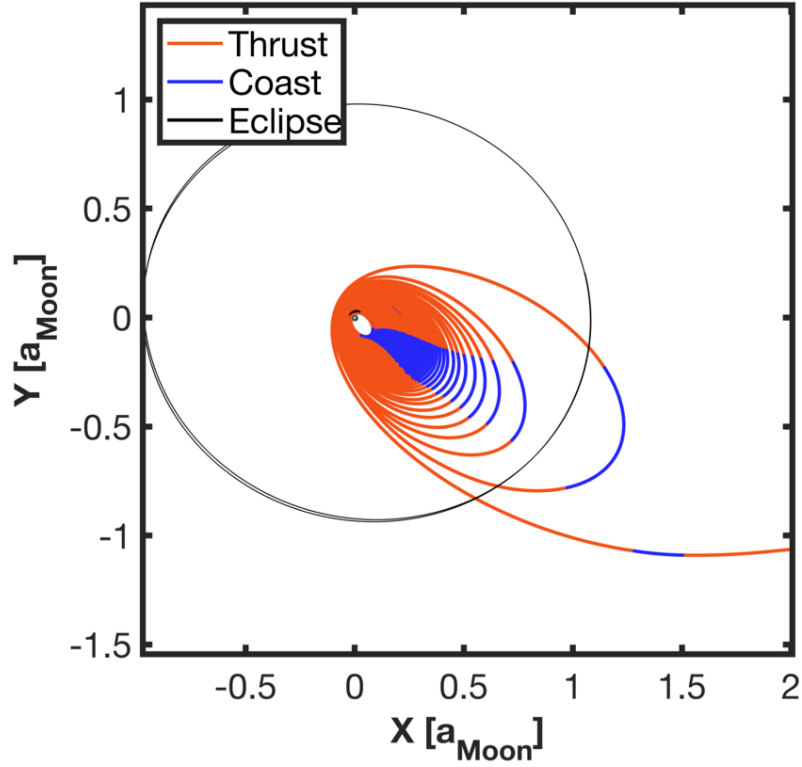


Figure 6.5: Families of low-thrust Moon-Moon transfers to Comet 45P identified by MBH. Transfers are shown for $v_{\infty 0} = 0.5, 0.6$, and 0.7 km/s.

the exception being the $v_{\infty 0} = 0.7$ km/s single swingby case. For some TOF ranges, the $v_{\infty 0} = 0.6$ km/s case actually performed better, despite the lower escape C_3 . This occurred because the higher $v_{\infty 0}$ can yield a less favorable spiral orientation that required more thrusting to transfer from GTO. Figure 6.8 demonstrates this, as the $v_{\infty 0} = 0.7$ km/s spiral's apoapsis extends farther beyond the Moon's orbit.

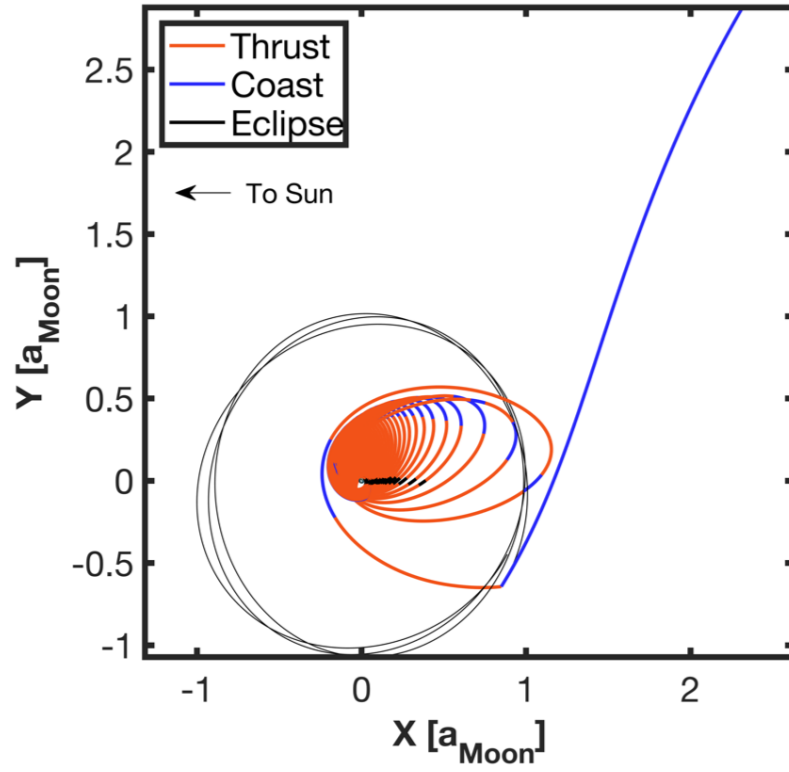


(a) Double Lunar gravity assist interplanetary phase.

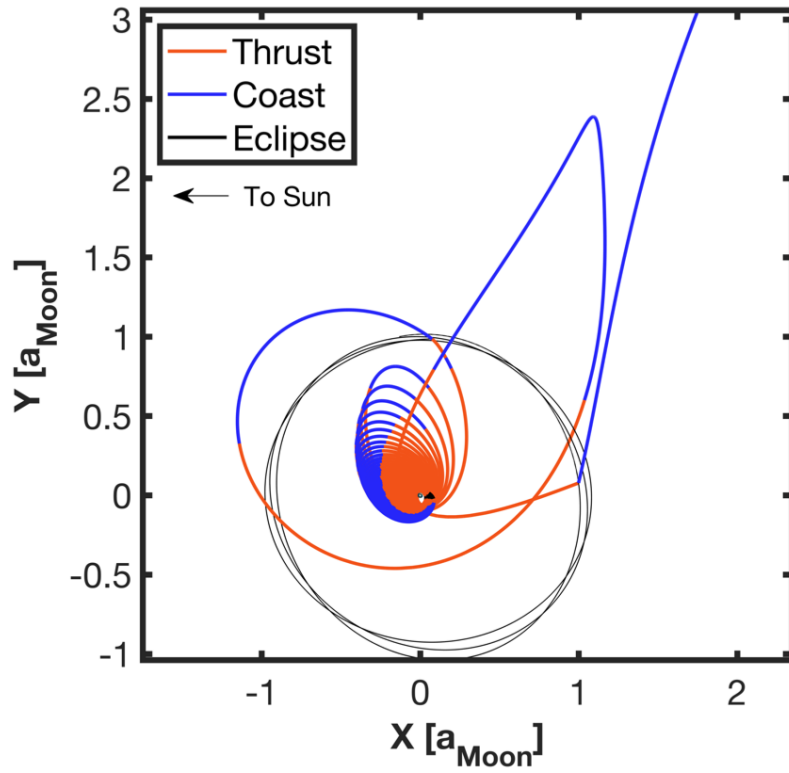


(b) Spiral escape with $C_3 = 0$ in ECI frame.

Figure 6.6: Sample escape trajectories to Comet 45P.



(c) Single Lunar gravity assist with $v_{\infty_0} = 0.7$ km/s.



(d) Double Lunar gravity assist with $v_{\infty_0} = 0.7$ km/s.

Figure 6.6: Sample escape trajectories to Comet 45P.

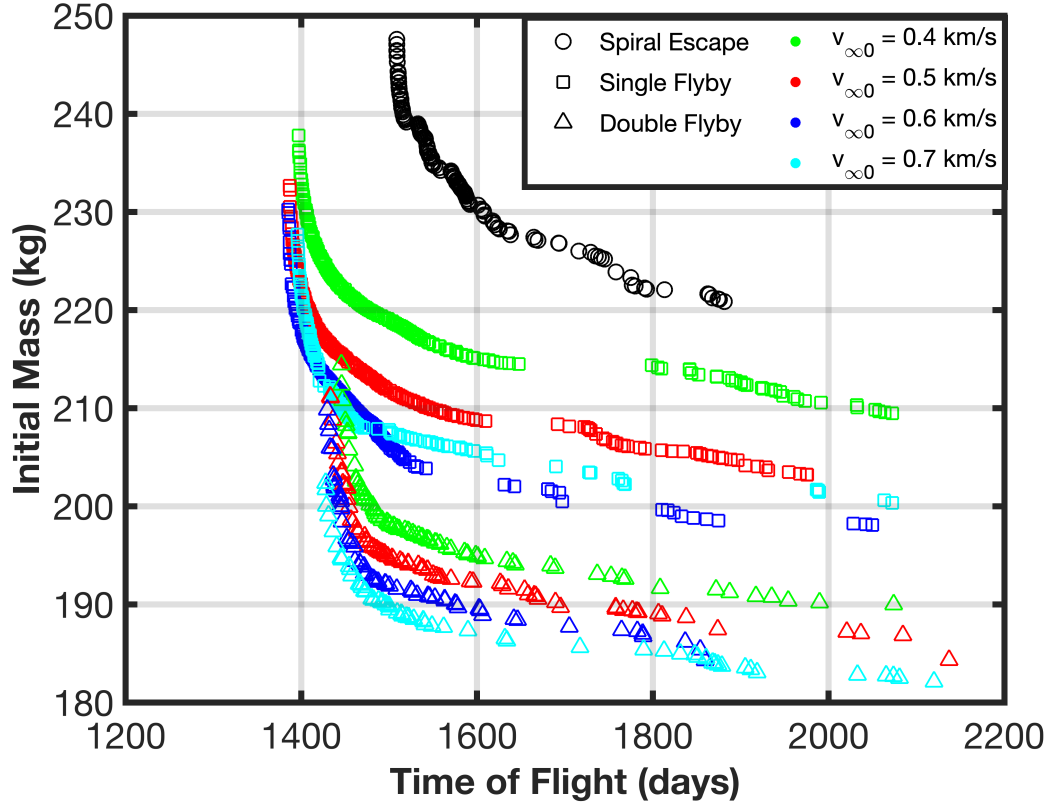
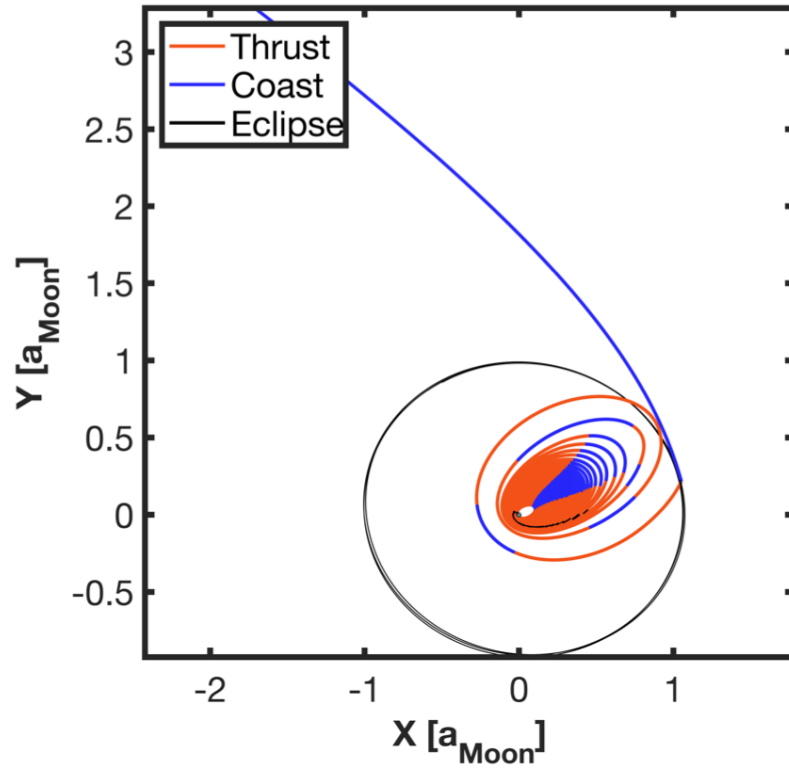
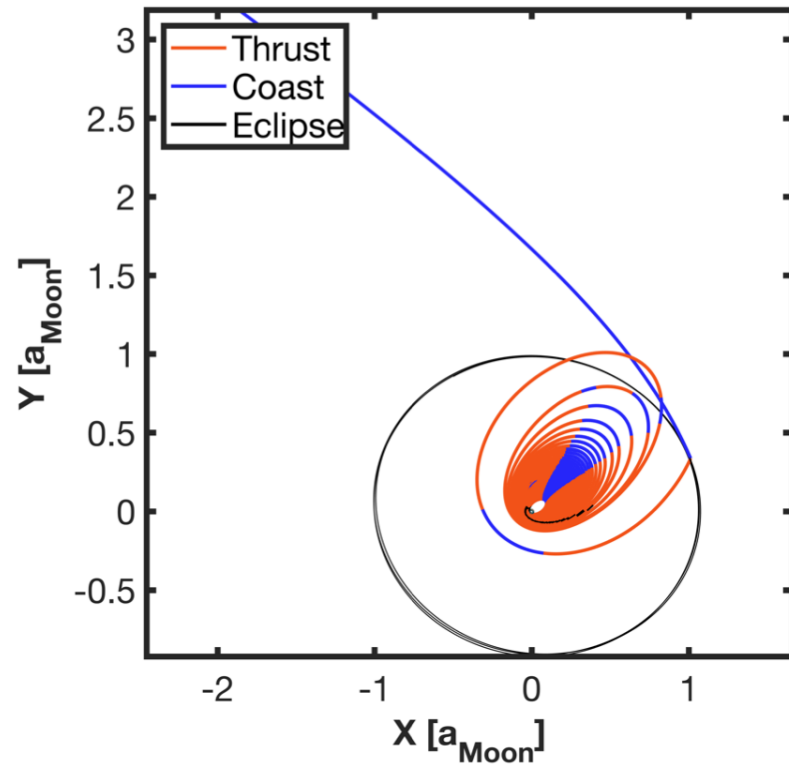


Figure 6.7: Comet 45P flyby mass trade for the spiral escape, single Lunar gravity assist, and double Lunar gravity assist cases.

Tisserand plots were introduced by Strange and Lunguski as a graphical tool for interplanetary gravity assist tour design [88]. The method uses Tisserand's criterion, shown in Equation 6.9, to identify contours of constant v_∞ that show how the spacecraft's orbit can change due to gravity assists. Figure 6.9 shows an energy Tisserand plot for the Earth-Moon system with the double swingby states included. The gravity assists do not move the spacecraft state exactly along the v_∞ contour because the contours were generated under the approximations that the planetary body is in a circular orbit and the spacecraft lies in the same plane as the perturbing body. The Moon's slight eccentricity and the out-of-plane component of the Moon-



(a) $v_{\infty_0} = 0.6$ km/s.



(b) $v_{\infty_0} = 0.7$ km/s.

Figure 6.8: Single gravity assist solutions in the ECI frame.

Moon legs violate those approximations, moving the spacecraft off the constant v_∞ line. Additionally, the Solar gravity perturbations and low-thrust VILMs provide the jump between the two flyby's, enabling the second swingby to achieve a higher escape energy.

$$T = \frac{r_{planet}}{a} + 2\sqrt{\frac{a(1-e^2)}{r_{planet}}} \cos i \quad (6.9)$$

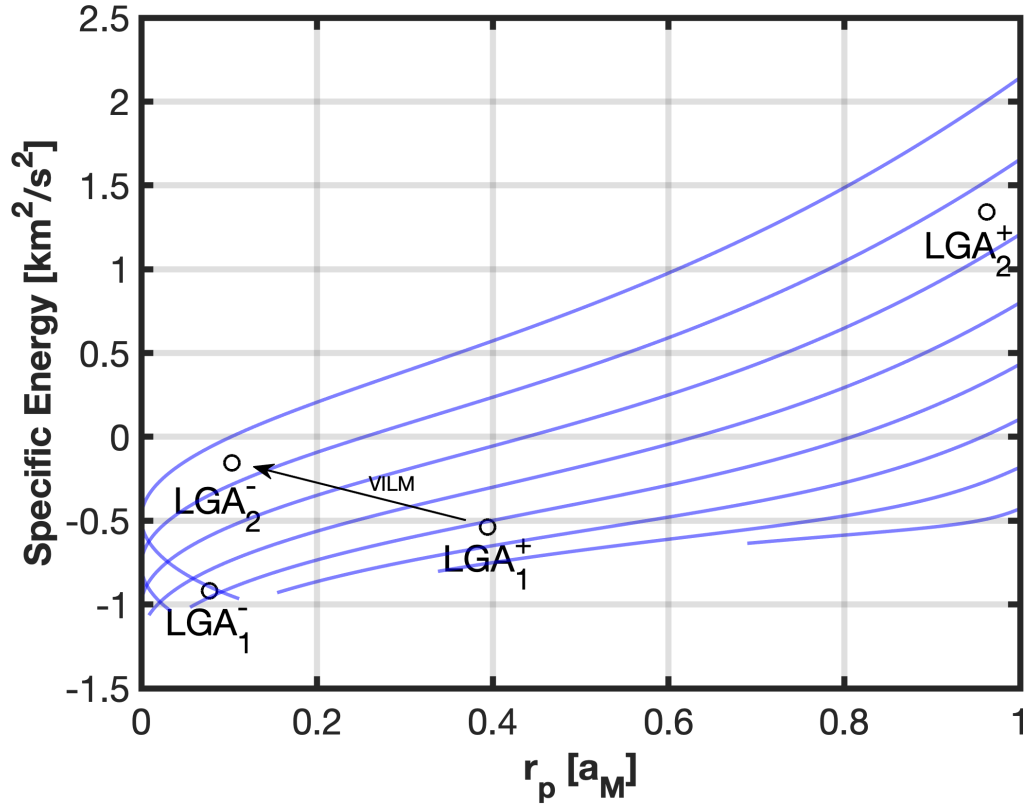


Figure 6.9: Earth-centered $E-r_p$ Tisserand plot with before and after flyby states included. LGA_1 and LGA_2 refer to the first and second Lunar gravity assists, respectively.

Chapter 7: Derivation and Application of Analytical Partial Derivatives of the Q-Law Thrust Vector

This chapter derives and applies the partial derivatives of the Q-Law thrust vector for a novel Q-Law shooting formulation as presented in Reference [89, 90]. Typically, Q-Law has been used within an evolutionary algorithm to stochastically vary the element gains to achieve more optimal solutions [29, 32, 34, 58, 65]. Although one Q-Law execution is very efficient, thousands of evaluations within an evolutionary algorithm will add up to considerable computation time. Gradient-based optimization methods typically require fewer function evaluations than evolutionary algorithms but converge to a locally optimal solution. This work extended the Q-Law gain tuning effort to gradient-based optimization by leveraging nonlinear programming to search for locally optimal Q-Law gain combinations and to enforce nonlinear constraints on the initial state. This was achieved by formulating the trajectory optimization as a Q-Law shooting problem. Additionally, the Q-Law shooting NLP was extended to include the well-known Sims-Flanagan interplanetary model [81]. Combining these two methods enabled the direct design of interplanetary spiral escape/capture missions as the cruise and spiral phases were designed in the same optimization problem.

As discussed in Section 4.3, NLP solvers can estimate the derivatives needed for each optimization step through finite differencing, but this approach is slow and can lead to derivative inaccuracy due to numerical truncation error, resulting in the solver taking sub-optimal steps during its line search. To overcome this, analytical partial derivatives are highly sought after in spacecraft trajectory optimization. When available and correctly implemented, they can have significant effects on the efficiency and accuracy of the solver [91, 92]. This chapter presents the derivation of the Q-Law thrust vector partial derivatives with respect to the gains and current spacecraft state and their application to trajectory optimization. These derivatives are required to generate a state transition matrix (STM) that contains the constraint sensitivities for the NLP solver. Matlab’s symbolic toolbox was used to generate some derivative expressions, and all analytical derivatives were verified using complex-step and dual numbers.

To demonstrate the capabilities of combining Q-Law with gradient-based optimization and the effectiveness of the analytical partial derivatives, several example problems were setup and solved. First, a Q-Law multiple shooting problem was formulated and tested on a common GTO-GEO transfer problem. The shooting problem was solved with both analytical derivatives and finite differencing as well as a variable number of trajectory phases to evaluate the effects on cost function and convergence rate. The logistic eclipse model detailed in Section 2.3 was implemented to provide a smooth power transition through the Earth’s shadow [21]. A logistic coasting function was also developed to limit the spacecraft thrust available at user-defined true anomaly ranges, providing a smooth, analytical coasting

mechanism that can be used within the NLP. Next, Q-Law shooting was used to aid in the design of low-thrust transfers to the Moon. Shannon *et al.* introduced the use of backward propagated Q-Law to rapidly design spiral transfers to the Moon, and they used a multi-objective evolutionary algorithm (MOEA) to reduce mass and epoch errors after the backward propagation [58, 59]. In this work, it is shown that Q-Law shooting improves upon this approach and can enforce mass and epoch boundary constraints to within NLP tolerance. This added capability enables Q-Law to exactly target a terminal state, mass, and epoch and results in a higher fidelity trajectory to the Moon. In addition to Q-Law-only transfers inside the NLP, the Q-Law shooting method can be extended to combine with other NLP transcription methods. For example, Q-Law shooting was combined with the Sims-Flanagan interplanetary model [81], which resulted in a single optimization problem that encompassed both the spiral escape/capture and the interplanetary phases. This represents a significant advancement in end-to-end trajectory design as the common preliminary design approach is to use Edelbaum’s equation, which limits the spirals to circular orbits and does not include eclipsing or coast arcs [93]. Using Q-Law shooting for the spiral phases yields more accurate flight time and propellant requirement estimates, and it lets the optimizer vary the design variables in an effort to minimize a global cost function for the full trajectory. These capabilities were demonstrated on a Mars transfer and capture spiral trajectory and a Lunar swingby escape spiral trajectory to Comet 45P. In each example problem, SNOPT was used to solve the NLP [50].

7.1 Computation of the Q-Law Thrust Vector Partial Derivatives

First the partials derivatives of $\frac{\partial Q}{\partial \mathbf{e}}$ with respect to the gains are needed. Taking the partials of Equation 3.24:

$$\frac{\partial^2 Q}{\partial \mathbf{e} \partial W_a} = \frac{\partial}{\partial W_a} \frac{\partial P}{\partial \mathbf{e}} \mathbf{W}^T \mathbf{V} + \frac{\partial}{\partial W_a} (1 + W_P P) \mathbf{W}^T \frac{\partial \mathbf{V}}{\partial \mathbf{e}} \quad (7.1)$$

$$\frac{\partial^2 Q}{\partial \mathbf{e} \partial W_a} = \frac{\partial P}{\partial \mathbf{e}} \frac{\partial \mathbf{W}^T}{\partial W_a} \mathbf{V} + (1 + W_P P) \frac{\partial \mathbf{W}^T}{\partial W_a} \frac{\partial \mathbf{V}}{\partial \mathbf{e}} \quad (7.2)$$

$$\frac{\partial^2 Q}{\partial \mathbf{e} \partial W_a} = \frac{\partial P}{\partial \mathbf{e}} [1, 0, 0, 0, 0] \mathbf{V} + (1 + W_P P) [1, 0, 0, 0, 0] \frac{\partial \mathbf{V}}{\partial \mathbf{e}} \quad (7.3)$$

$$\frac{\partial^2 Q}{\partial \mathbf{e} \partial W_a} = \frac{\partial P}{\partial \mathbf{e}} \mathbf{S}_a \left[\frac{d(\mathbf{a}, \mathbf{a}_T)}{\dot{\mathbf{a}}_{xx}} \right]^2 + (1 + W_P P) \frac{\partial V_a}{\partial \mathbf{e}} \quad (7.4)$$

The partials derivatives with respect to the other gains are derived in a similar fashion.

$$\frac{\partial^2 Q}{\partial \mathbf{e} \partial W_e} = \frac{\partial P}{\partial \mathbf{e}} \mathbf{S}_e \left[\frac{d(\mathbf{e}, \mathbf{e}_T)}{\dot{\mathbf{e}}_{xx}} \right]^2 + (1 + W_P P) \frac{\partial V_e}{\partial \mathbf{e}} \quad (7.5)$$

$$\frac{\partial^2 Q}{\partial \mathbf{e} \partial W_i} = \frac{\partial P}{\partial \mathbf{e}} \mathbf{S}_i \left[\frac{d(\mathbf{i}, \mathbf{i}_T)}{\dot{\mathbf{i}}_{xx}} \right]^2 + (1 + W_P P) \frac{\partial V_i}{\partial \mathbf{e}} \quad (7.6)$$

$$\frac{\partial^2 Q}{\partial \mathbf{e} \partial W_\omega} = \frac{\partial P}{\partial \mathbf{e}} \mathbf{S}_\omega \left[\frac{d(\omega, \omega_T)}{\dot{\omega}_{xx}} \right]^2 + (1 + W_P P) \frac{\partial V_\omega}{\partial \mathbf{e}} \quad (7.7)$$

$$\frac{\partial^2 Q}{\partial \mathbf{c} \partial W_\Omega} = \frac{\partial P}{\partial \mathbf{c}} \mathbf{S}_\Omega \left[\frac{d(\Omega, \Omega_T)}{\dot{\Omega}_{xx}} \right]^2 + (1 + W_P P) \frac{\partial V_\Omega}{\partial \mathbf{c}} \quad (7.8)$$

Now, using $\frac{\partial^2 Q}{\partial \mathbf{c} \partial W_\Omega}$, the D_1 , D_2 , and D_3 coefficient derivatives can be found, leading to the thrust vector gain partial derivatives.

7.1.1 Partial Derivatives With Respect to W_a

$$\frac{\partial D_1}{\partial W_a} = \frac{\partial^2 Q}{\partial \mathbf{c} \partial W_a} \frac{\partial \mathbf{c}}{\partial f_\theta} \quad (7.9)$$

$$\frac{\partial D_2}{\partial W_a} = \frac{\partial^2 Q}{\partial \mathbf{c} \partial W_a} \frac{\partial \mathbf{c}}{\partial f_r} \quad (7.10)$$

$$\frac{\partial D_3}{\partial W_a} = \frac{\partial^2 Q}{\partial \mathbf{c} \partial W_a} \frac{\partial \mathbf{c}}{\partial f_h} \quad (7.11)$$

$$\frac{\partial u_r}{\partial W_a} = \frac{D_2(D_1 \frac{\partial D_1}{\partial W_a} + D_2 \frac{\partial D_2}{\partial W_a} + D_3 \frac{\partial D_3}{\partial W_a})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial W_a}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.12)$$

$$\frac{\partial u_\theta}{\partial W_a} = \frac{D_1(D_1 \frac{\partial D_1}{\partial W_a} + D_2 \frac{\partial D_2}{\partial W_a} + D_3 \frac{\partial D_3}{\partial W_a})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial W_a}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.13)$$

$$\frac{\partial u_h}{\partial W_a} = \frac{D_3(D_1 \frac{\partial D_1}{\partial W_a} + D_2 \frac{\partial D_2}{\partial W_a} + D_3 \frac{\partial D_3}{\partial W_a})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial W_a}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.14)$$

7.1.2 Partial Derivatives With Respect to W_e

$$\frac{\partial D_1}{\partial W_e} = \frac{\partial^2 Q}{\partial \alpha \partial W_e} \frac{\partial \dot{\alpha}}{\partial f_\theta} \quad (7.15)$$

$$\frac{\partial D_2}{\partial W_e} = \frac{\partial^2 Q}{\partial \alpha \partial W_e} \frac{\partial \dot{\alpha}}{\partial f_r} \quad (7.16)$$

$$\frac{\partial D_3}{\partial W_e} = \frac{\partial^2 Q}{\partial \alpha \partial W_e} \frac{\partial \dot{\alpha}}{\partial f_h} \quad (7.17)$$

$$\frac{\partial u_r}{\partial W_e} = \frac{D_2(D_1 \frac{\partial D_1}{\partial W_e} + D_2 \frac{\partial D_2}{\partial W_e} + D_3 \frac{\partial D_3}{\partial W_e})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial W_e}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.18)$$

$$\frac{\partial u_\theta}{\partial W_e} = \frac{D_1(D_1 \frac{\partial D_1}{\partial W_e} + D_2 \frac{\partial D_2}{\partial W_e} + D_3 \frac{\partial D_3}{\partial W_e})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial W_e}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.19)$$

$$\frac{\partial u_h}{\partial W_e} = \frac{D_3(D_1 \frac{\partial D_1}{\partial W_e} + D_2 \frac{\partial D_2}{\partial W_e} + D_3 \frac{\partial D_3}{\partial W_e})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial W_e}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.20)$$

7.1.3 Partial Derivatives With Respect to W_i

$$\frac{\partial D_1}{\partial W_i} = \frac{\partial^2 Q}{\partial \alpha \partial W_i} \frac{\partial \dot{\alpha}}{\partial f_\theta} \quad (7.21)$$

$$\frac{\partial D_2}{\partial W_i} = \frac{\partial^2 Q}{\partial \alpha \partial W_i} \frac{\partial \dot{\alpha}}{\partial f_r} \quad (7.22)$$

$$\frac{\partial D_3}{\partial W_i} = \frac{\partial^2 Q}{\partial \alpha \partial W_i} \frac{\partial \dot{\alpha}}{\partial f_h} \quad (7.23)$$

$$\frac{\partial u_r}{\partial W_i} = \frac{D_2(D_1 \frac{\partial D_1}{\partial W_i} + D_2 \frac{\partial D_2}{\partial W_i} + D_3 \frac{\partial D_3}{\partial W_i})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial W_i}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.24)$$

$$\frac{\partial u_\theta}{\partial W_i} = \frac{D_1(D_1 \frac{\partial D_1}{\partial W_i} + D_2 \frac{\partial D_2}{\partial W_i} + D_3 \frac{\partial D_3}{\partial W_i})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial W_i}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.25)$$

$$\frac{\partial u_h}{\partial W_i} = \frac{D_3(D_1 \frac{\partial D_1}{\partial W_i} + D_2 \frac{\partial D_2}{\partial W_i} + D_3 \frac{\partial D_3}{\partial W_i})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial W_i}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.26)$$

7.1.4 Partial Derivatives With Respect to W_ω

$$\frac{\partial D_1}{\partial W_\omega} = \frac{\partial^2 Q}{\partial \alpha \partial W_\omega} \frac{\partial \dot{\alpha}}{\partial f_\theta} \quad (7.27)$$

$$\frac{\partial D_2}{\partial W_\omega} = \frac{\partial^2 Q}{\partial \alpha \partial W_\omega} \frac{\partial \dot{\alpha}}{\partial f_r} \quad (7.28)$$

$$\frac{\partial D_3}{\partial W_\omega} = \frac{\partial^2 Q}{\partial \alpha \partial W_\omega} \frac{\partial \dot{\alpha}}{\partial f_h} \quad (7.29)$$

$$\frac{\partial u_r}{\partial W_\omega} = \frac{D_2(D_1 \frac{\partial D_1}{\partial W_\omega} + D_2 \frac{\partial D_2}{\partial W_\omega} + D_3 \frac{\partial D_3}{\partial W_\omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial W_\omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.30)$$

$$\frac{\partial u_\theta}{\partial W_\omega} = \frac{D_1(D_1 \frac{\partial D_1}{\partial W_\omega} + D_2 \frac{\partial D_2}{\partial W_\omega} + D_3 \frac{\partial D_3}{\partial W_\omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial W_\omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.31)$$

$$\frac{\partial u_h}{\partial W_\omega} = \frac{D_3(D_1 \frac{\partial D_1}{\partial W_\omega} + D_2 \frac{\partial D_2}{\partial W_\omega} + D_3 \frac{\partial D_3}{\partial W_\omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial W_\omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.32)$$

7.1.5 Partial Derivatives With Respect to W_Ω

$$\frac{\partial D_1}{\partial W_\Omega} = \frac{\partial^2 Q}{\partial \alpha \partial W_\Omega} \frac{\partial \alpha}{\partial f_\theta} \quad (7.33)$$

$$\frac{\partial D_2}{\partial W_\Omega} = \frac{\partial^2 Q}{\partial \alpha \partial W_\Omega} \frac{\partial \alpha}{\partial f_r} \quad (7.34)$$

$$\frac{\partial D_3}{\partial W_\Omega} = \frac{\partial^2 Q}{\partial \alpha \partial W_\Omega} \frac{\partial \alpha}{\partial f_h} \quad (7.35)$$

$$\frac{\partial u_r}{\partial W_\Omega} = \frac{D_2(D_1 \frac{\partial D_1}{\partial W_\Omega} + D_2 \frac{\partial D_2}{\partial W_\Omega} + D_3 \frac{\partial D_3}{\partial W_\Omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial W_\Omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.36)$$

$$\frac{\partial u_\theta}{\partial W_\Omega} = \frac{D_1(D_1 \frac{\partial D_1}{\partial W_\Omega} + D_2 \frac{\partial D_2}{\partial W_\Omega} + D_3 \frac{\partial D_3}{\partial W_\Omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial W_\Omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.37)$$

$$\frac{\partial u_h}{\partial W_\Omega} = \frac{D_3(D_1 \frac{\partial D_1}{\partial W_\Omega} + D_2 \frac{\partial D_2}{\partial W_\Omega} + D_3 \frac{\partial D_3}{\partial W_\Omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial W_\Omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.38)$$

Next, the thrust vector partial derivatives with respect to the spacecraft state are found. First, the partial derivatives of Equations 3.20 to 3.22 are taken. In this step, symbolic derivatives were used to find expressions for the Hessian elements of Q .

7.1.6 Partial Derivatives With Respect to Semi-Major Axis

$$\frac{\partial D_1}{\partial a} = \frac{\partial^2 Q}{\partial \epsilon \partial a} \frac{\partial \dot{\epsilon}}{\partial f_\theta} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_\theta \partial a} \quad (7.39)$$

$$\frac{\partial D_2}{\partial a} = \frac{\partial^2 Q}{\partial \epsilon \partial a} \frac{\partial \dot{\epsilon}}{\partial f_r} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_r \partial a} \quad (7.40)$$

$$\frac{\partial D_3}{\partial a} = \frac{\partial^2 Q}{\partial \epsilon \partial a} \frac{\partial \dot{\epsilon}}{\partial f_h} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_h \partial a} \quad (7.41)$$

$$\frac{\partial u_r}{\partial a} = \frac{D_2(D_1 \frac{\partial D_1}{\partial a} + D_2 \frac{\partial D_2}{\partial a} + D_3 \frac{\partial D_3}{\partial a})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial a}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.42)$$

$$\frac{\partial u_\theta}{\partial a} = \frac{D_1(D_1 \frac{\partial D_1}{\partial a} + D_2 \frac{\partial D_2}{\partial a} + D_3 \frac{\partial D_3}{\partial a})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial a}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.43)$$

$$\frac{\partial u_h}{\partial a} = \frac{D_3(D_1 \frac{\partial D_1}{\partial a} + D_2 \frac{\partial D_2}{\partial a} + D_3 \frac{\partial D_3}{\partial a})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial a}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.44)$$

7.1.7 Partial Derivatives With Respect to Eccentricity

$$\frac{\partial D_1}{\partial e} = \frac{\partial^2 Q}{\partial \alpha \partial e} \frac{\partial \dot{\alpha}}{\partial f_\theta} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_\theta \partial e} \quad (7.45)$$

$$\frac{\partial D_2}{\partial e} = \frac{\partial^2 Q}{\partial \alpha \partial e} \frac{\partial \dot{\alpha}}{\partial f_r} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_r \partial e} \quad (7.46)$$

$$\frac{\partial D_3}{\partial e} = \frac{\partial^2 Q}{\partial \alpha \partial e} \frac{\partial \dot{\alpha}}{\partial f_h} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_h \partial e} \quad (7.47)$$

$$\frac{\partial u_r}{\partial e} = \frac{D_2(D_1 \frac{\partial D_1}{\partial e} + D_2 \frac{\partial D_2}{\partial e} + D_3 \frac{\partial D_3}{\partial e})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial e}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.48)$$

$$\frac{\partial u_\theta}{\partial e} = \frac{D_1(D_1 \frac{\partial D_1}{\partial e} + D_2 \frac{\partial D_2}{\partial e} + D_3 \frac{\partial D_3}{\partial e})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial e}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.49)$$

$$\frac{\partial u_h}{\partial e} = \frac{D_3(D_1 \frac{\partial D_1}{\partial e} + D_2 \frac{\partial D_2}{\partial e} + D_3 \frac{\partial D_3}{\partial e})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial e}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.50)$$

7.1.8 Partial Derivatives With Respect to Inclination

$$\frac{\partial D_1}{\partial i} = \frac{\partial^2 Q}{\partial \alpha \partial i} \frac{\partial \dot{\alpha}}{\partial f_\theta} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_\theta \partial i} \quad (7.51)$$

$$\frac{\partial D_2}{\partial i} = \frac{\partial^2 Q}{\partial \epsilon \partial i} \frac{\partial \dot{\epsilon}}{\partial f_r} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_r \partial i} \quad (7.52)$$

$$\frac{\partial D_3}{\partial i} = \frac{\partial^2 Q}{\partial \epsilon \partial i} \frac{\partial \dot{\epsilon}}{\partial f_h} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_h \partial i} \quad (7.53)$$

$$\frac{\partial u_r}{\partial i} = \frac{D_2(D_1 \frac{\partial D_1}{\partial i} + D_2 \frac{\partial D_2}{\partial i} + D_3 \frac{\partial D_3}{\partial i})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial i}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.54)$$

$$\frac{\partial u_\theta}{\partial i} = \frac{D_1(D_1 \frac{\partial D_1}{\partial i} + D_2 \frac{\partial D_2}{\partial i} + D_3 \frac{\partial D_3}{\partial i})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial i}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.55)$$

$$\frac{\partial u_h}{\partial i} = \frac{D_3(D_1 \frac{\partial D_1}{\partial i} + D_2 \frac{\partial D_2}{\partial i} + D_3 \frac{\partial D_3}{\partial i})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial i}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.56)$$

7.1.9 Partial Derivatives With Respect to Argument of Periapsis

$$\frac{\partial D_1}{\partial \omega} = \frac{\partial^2 Q}{\partial \epsilon \partial \omega} \frac{\partial \dot{\epsilon}}{\partial f_\theta} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_\theta \partial \omega} \quad (7.57)$$

$$\frac{\partial D_2}{\partial \omega} = \frac{\partial^2 Q}{\partial \epsilon \partial \omega} \frac{\partial \dot{\epsilon}}{\partial f_r} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_r \partial \omega} \quad (7.58)$$

$$\frac{\partial D_3}{\partial \omega} = \frac{\partial^2 Q}{\partial \epsilon \partial \omega} \frac{\partial \dot{\epsilon}}{\partial f_h} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_h \partial \omega} \quad (7.59)$$

$$\frac{\partial u_r}{\partial \omega} = \frac{D_2(D_1 \frac{\partial D_1}{\partial \omega} + D_2 \frac{\partial D_2}{\partial \omega} + D_3 \frac{\partial D_3}{\partial \omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial \omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.60)$$

$$\frac{\partial u_\theta}{\partial \omega} = \frac{D_1(D_1 \frac{\partial D_1}{\partial \omega} + D_2 \frac{\partial D_2}{\partial \omega} + D_3 \frac{\partial D_3}{\partial \omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial \omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.61)$$

$$\frac{\partial u_h}{\partial \omega} = \frac{D_3(D_1 \frac{\partial D_1}{\partial \omega} + D_2 \frac{\partial D_2}{\partial \omega} + D_3 \frac{\partial D_3}{\partial \omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial \omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.62)$$

7.1.10 Partial Derivatives With Respect to Longitude of Ascending Node

$$\frac{\partial D_1}{\partial \Omega} = \frac{\partial^2 Q}{\partial \epsilon \partial \Omega} \frac{\partial \dot{\epsilon}}{\partial f_\theta} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_\theta \partial \Omega} \quad (7.63)$$

$$\frac{\partial D_2}{\partial \Omega} = \frac{\partial^2 Q}{\partial \epsilon \partial \Omega} \frac{\partial \dot{\epsilon}}{\partial f_r} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_r \partial \Omega} \quad (7.64)$$

$$\frac{\partial D_3}{\partial \Omega} = \frac{\partial^2 Q}{\partial \epsilon \partial \Omega} \frac{\partial \dot{\epsilon}}{\partial f_h} + \frac{\partial Q}{\partial \epsilon} \frac{\partial^2 \dot{\epsilon}}{\partial f_h \partial \Omega} \quad (7.65)$$

$$\frac{\partial u_r}{\partial \Omega} = \frac{D_2(D_1 \frac{\partial D_1}{\partial \Omega} + D_2 \frac{\partial D_2}{\partial \Omega} + D_3 \frac{\partial D_3}{\partial \Omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial \Omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.66)$$

$$\frac{\partial u_\theta}{\partial \Omega} = \frac{D_1(D_1 \frac{\partial D_1}{\partial \Omega} + D_2 \frac{\partial D_2}{\partial \Omega} + D_3 \frac{\partial D_3}{\partial \Omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial \Omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.67)$$

$$\frac{\partial u_h}{\partial \Omega} = \frac{D_3(D_1 \frac{\partial D_1}{\partial \Omega} + D_2 \frac{\partial D_2}{\partial \Omega} + D_3 \frac{\partial D_3}{\partial \Omega})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial \Omega}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.68)$$

7.1.11 Partial Derivatives With Respect to True Anomaly

$$\frac{\partial D_1}{\partial \theta} = \frac{\partial^2 Q}{\partial \alpha \partial \theta} \frac{\partial \dot{\alpha}}{\partial f_\theta} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_\theta \partial \theta} \quad (7.69)$$

$$\frac{\partial D_2}{\partial \theta} = \frac{\partial^2 Q}{\partial \alpha \partial \theta} \frac{\partial \dot{\alpha}}{\partial f_r} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_r \partial \theta} \quad (7.70)$$

$$\frac{\partial D_3}{\partial \theta} = \frac{\partial^2 Q}{\partial \alpha \partial \theta} \frac{\partial \dot{\alpha}}{\partial f_h} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_h \partial \theta} \quad (7.71)$$

$$\frac{\partial u_r}{\partial \theta} = \frac{D_2(D_1 \frac{\partial D_1}{\partial \theta} + D_2 \frac{\partial D_2}{\partial \theta} + D_3 \frac{\partial D_3}{\partial \theta})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial \theta}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.72)$$

$$\frac{\partial u_\theta}{\partial \theta} = \frac{D_1(D_1 \frac{\partial D_1}{\partial \theta} + D_2 \frac{\partial D_2}{\partial \theta} + D_3 \frac{\partial D_3}{\partial \theta})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial \theta}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.73)$$

$$\frac{\partial u_h}{\partial \theta} = \frac{D_3(D_1 \frac{\partial D_1}{\partial \theta} + D_2 \frac{\partial D_2}{\partial \theta} + D_3 \frac{\partial D_3}{\partial \theta})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial \theta}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.74)$$

7.1.12 Partial Derivatives With Respect to Mass

$$\frac{\partial D_1}{\partial m} = \frac{\partial^2 Q}{\partial \alpha \partial m} \frac{\partial \dot{\alpha}}{\partial f_\theta} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_\theta \partial m} \quad (7.75)$$

$$\frac{\partial D_2}{\partial m} = \frac{\partial^2 Q}{\partial \alpha \partial m} \frac{\partial \dot{\alpha}}{\partial f_r} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_r \partial m} \quad (7.76)$$

$$\frac{\partial D_3}{\partial m} = \frac{\partial^2 Q}{\partial \alpha \partial m} \frac{\partial \dot{\alpha}}{\partial f_h} + \frac{\partial Q}{\partial \alpha} \frac{\partial^2 \dot{\alpha}}{\partial f_h \partial m} \quad (7.77)$$

$$\frac{\partial u_r}{\partial m} = \frac{D_2(D_1 \frac{\partial D_1}{\partial m} + D_2 \frac{\partial D_2}{\partial m} + D_3 \frac{\partial D_3}{\partial m})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_2}{\partial m}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.78)$$

$$\frac{\partial u_\theta}{\partial m} = \frac{D_1(D_1 \frac{\partial D_1}{\partial m} + D_2 \frac{\partial D_2}{\partial m} + D_3 \frac{\partial D_3}{\partial m})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_1}{\partial m}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.79)$$

$$\frac{\partial u_h}{\partial m} = \frac{D_3(D_1 \frac{\partial D_1}{\partial m} + D_2 \frac{\partial D_2}{\partial m} + D_3 \frac{\partial D_3}{\partial m})}{(D_1^2 + D_2^2 + D_3^2)^{3/2}} - \frac{\frac{\partial D_3}{\partial m}}{\sqrt{D_1^2 + D_2^2 + D_3^2}} \quad (7.80)$$

7.2 Shooting Methods

Shooting algorithms use a simulation of the system to evaluate the initial conditions and control variables. The trajectory segment(s) is repeatedly propagated, and after each evaluation, a differential correction scheme or optimizer adjusts the initial conditions and control variables to satisfy a set of constraints. If used within an NLP, a cost function containing the state and/or control variables will be minimized as well. The propagate-and-update process is repeated until the constraints are satisfied within a specified tolerance and the cost function is minimized.

The most basic form of the shooting architecture is the single shooting prob-

lem, which is shown in Figure 7.1. In this problem, only one trajectory arc is present, and the initial conditions and/or control variables are adjusted to directly evaluate their effect on the final state after propagating a duration of time T . The constraint $\mathbf{x}(T) - \mathbf{x}_{Target} = \mathbf{0}$ is enforced to ensure the target state is hit, and the cost function $J(\mathbf{x}_0, \mathbf{u})$ is minimized.

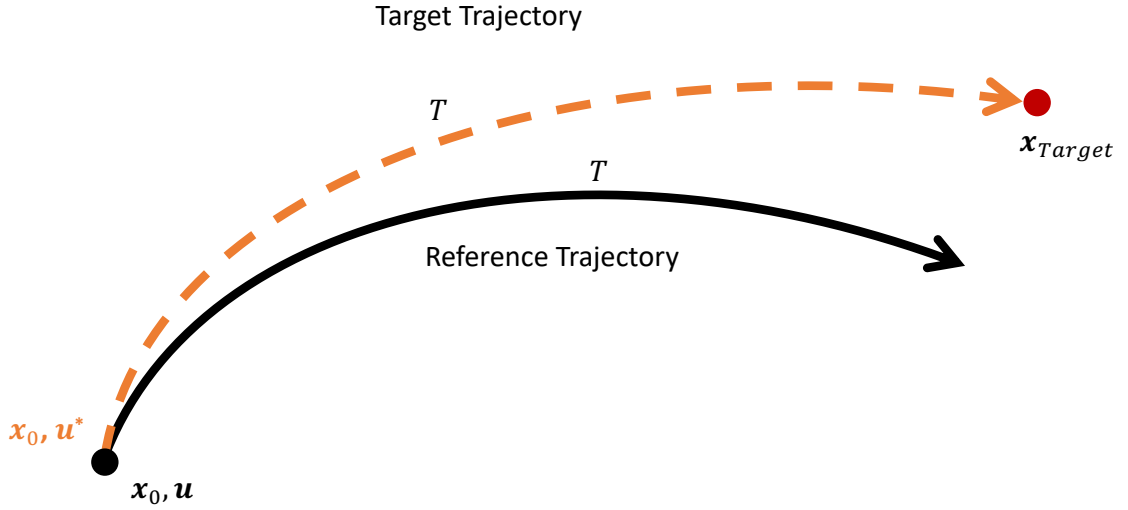


Figure 7.1: Single shooting.

The multiple shooting approach is similar to the single shooting approach, with the exception that there are multiple trajectory segments now present. By dividing the trajectory into multiple arcs, the problem sensitivity can be reduced. This is very beneficial when one or more phases of the trajectory occur in more sensitive dynamic regimes, such as a close pass to a planetary body. Subdividing the trajectory can localize these sensitivities to one segment rather than propagating the global initial conditions all the way through, as is done in single shooting. Additionally, solution accuracy can be improved. This occurs because numerical integration errors grow with time, so multiple subarcs with shorter propagation times reduce the overall

errors within the trajectory. A depiction of a multiple shooting scheme is shown in Figure 7.2. For an N segment problem, $\sum_{i=1}^{N-1} \mathbf{x}_i^f - \mathbf{x}_{i+1}^0 = \mathbf{0}$ is enforced as continuity constraints between the segments. The arrival constraint and cost function are defined the same as in the single shooting problem.

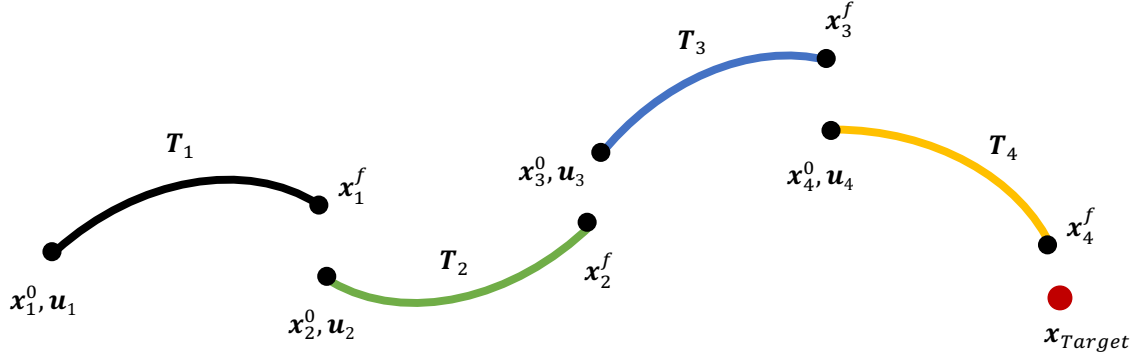


Figure 7.2: Multiple shooting.

7.3 Q-Law Shooting Setup

For a particular Q-Law trajectory, the control variables are the gains, W_{∞} , and the spacecraft state is defined as

$$\mathbf{x} = [a, e, i, \omega, \Omega, \theta, m]^T \quad (7.81)$$

When Q-Law was used within an NLP, the solver chooses the gains in order to optimize the flight time cost function, as shown in Equation 7.82. Additionally, constraints like Equation 7.83 were enforced to ensure the final state after integration satisfies the desired terminal boundary condition. To effectively optimize the objective function and satisfy the constraints, the NLP decision vector \mathbf{X} , shown in

Equation 7.84, includes the initial state, the element gains, and the mission flight time. The initial state in Equation 7.84 can represent the initial conditions for forward or backward propagation. Also, the decision vector can be augmented with additional gain combinations and intermediate states for the multiple shooting formulation.

$$\mathbf{J} = t_f \quad (7.82)$$

$$\mathbf{c} = \mathbf{x}_{des} - F(\mathbf{x}_0, \mathbf{W}, t_f) = 0 \quad (7.83)$$

$$\mathbf{X} = [a_0, e_0, i_0, \omega_0, \Omega_0, \theta_0, m_0, W_a, W_e, W_i, W_\omega, W_\Omega, t_f]^T \quad (7.84)$$

For an NLP solver to enforce the constraint in Equation 7.83 and minimize the cost in Equation 7.82, the sensitivities of the final state to the initial state, gains, and flight time were needed. These sensitivities make up the STM, Φ , which is defined as

$$\Phi = \frac{\partial \mathbf{X}_f}{\partial \mathbf{X}_0} = \left[\frac{\partial \mathbf{X}_f}{\partial \mathbf{x}_0}, \frac{\partial \mathbf{X}_f}{\partial \mathbf{W}}, \frac{\partial \mathbf{X}_f}{\partial t_f} \right] \quad (7.85)$$

and evolves according to Equations 7.86 through 7.90.

$$\dot{\Phi} = A\Phi \quad (7.86)$$

$$\Phi(t_0) = \mathbb{1} \quad (7.87)$$

$$A = \frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \quad (7.88)$$

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) \quad (7.89)$$

$$\mathbf{X}(t_0) = \mathbf{X}_0 \quad (7.90)$$

The STM was integrated alongside the trajectory to yield the final sensitivity matrix, which was reported to the NLP solver. However, the state dynamics and A matrix depend on the Q-Law thrust vector, $\mathbf{u} = (u_r, u_\theta, u_h)$, which in turn depends on the current state and gains. In order to evaluate the A matrix at each time step during the integration, the partial derivatives of the thrust vector with respect to the current state, $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$, and gains, $\frac{\partial \mathbf{u}}{\partial \mathbf{w}}$, were needed.

As demonstrated by Pellegrini and Russell [94], fixed-step integrators are most suitable when using the variational equations to generate a trajectory STM. A fixed-step is advantageous because the time step is chosen independent of the decision variables, whereas variable-step integrators adjust the step size according to error estimates dependent on the prior step. To accurately generate the STM, the partial derivatives of the time step with respect to the NLP decision variables are required, as shown for a generic explicit Runge-Kutta step in Equations 7.91 to 7.97. An RK

integration step with s stages is defined as

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \sum_{i=1}^s b_i \mathbf{k}_i \quad (7.91)$$

where b_i are the quadrature weights for the given method and

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(t_n, \mathbf{X}_n)h \\ \mathbf{k}_i &= \mathbf{f}(t_i + c_i h, \mathbf{X}_{n(i)})h \end{aligned} \quad (7.92)$$

In Equation 7.92, $\mathbf{X}_{n(i)}$ represents the state vector at the i^{th} stage of the current (n^{th}) integration step and is defined as

$$\mathbf{X}_{n(i)} = \mathbf{X}_n + \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j \quad (7.93)$$

The values for b_i , c_i , and a_{ij} can be obtained from the Butcher tableau for the chosen RK method. To determine the STM update equation for a given RK step, the partial derivatives of Equation 7.91 are taken with respect to the initial state.

$$\frac{\partial \mathbf{X}_{n+1}}{\partial \mathbf{X}_0} = \frac{\partial \mathbf{X}_n}{\partial \mathbf{X}_0} + \sum_{i=1}^s b_i \frac{\partial \mathbf{k}_i}{\partial \mathbf{X}_0} \quad (7.94)$$

Taking the partial derivatives of the definition in Equation 7.92 yields

$$\frac{\partial \mathbf{k}_i}{\partial \mathbf{X}_0} = \mathbf{f}(t_i + c_i h, \mathbf{X}_{n(i)}) \frac{\partial h}{\partial \mathbf{X}} \Big|_{(t_i + c_i h, \mathbf{X}_{n(i)})} + h \frac{\partial \mathbf{f}}{\partial \mathbf{X}} \Big|_{(t_i + c_i h, \mathbf{X}_{n(i)})} \frac{\partial \mathbf{X}_{n(i)}}{\partial \mathbf{X}_0} \quad (7.95)$$

$$\frac{\partial \mathbf{k}_i}{\partial \mathbf{X}_0} = \left(\mathbf{f}(t_i + c_i h, \mathbf{X}_{n(i)}) \frac{\partial h}{\partial \mathbf{X}} \Big|_{(t_i + c_i h, \mathbf{X}_{n(i)})} + h A|_{(t_i + c_i h, \mathbf{X}_{n(i)})} \right) \Phi_{n(i)} \quad (7.96)$$

with $\Phi_{n(i)}$ being the STM value at the i^{th} stage of the current (n^{th}) integration step. Substituting back into Equation 7.94 and using the definition in Equation 7.85 results in the STM update equation.

$$\Phi_{n+1} = \Phi_n + \sum_{i=1}^s b_i \left(\mathbf{f}(t_i + c_i h, \mathbf{X}_{n(i)}) \frac{\partial h}{\partial \mathbf{X}} \Big|_{(t_i + c_i h, \mathbf{X}_{n(i)})} + h A|_{(t_i + c_i h, \mathbf{X}_{n(i)})} \right) \Phi_{n(i)} \quad (7.97)$$

A nominal time step was used for all integration steps up until a final targeting step which matched the flight time decision variable exactly. As a result, $\frac{\partial h}{\partial \mathbf{X}} = [0, 0, \dots, 0]$ for all steps except the final integration step, where $h = t_f - t$ and $\frac{\partial h}{\partial \mathbf{X}} = [0, 0, \dots, 1]$.

The most basic Q-Law NLP formulation takes the form of a single shooting problem. Additional phases were introduced within the Q-Law trajectory for a multiple shooting formulation, with the NLP decision vector augmented to include the initial state and gains for each phase. The NLP solver then identified the optimal gains for each phase. Since more phases were introduced, solution optimality increased as more optimal gain combinations can be used for the varying dynamical environment. However, including additional phases resulted in a larger number of control variables and defect constraints between the terminal and initial states of two consecutive phases, thus increasing NLP size and complexity. In the multi-

ple shooting problem, a new STM was constructed for each phase and propagated alongside the spacecraft to determine the sensitivities of the resulting defect constraints to the phase initial state. The Q-Law multiple shooting concept is depicted in Figure 7.3.

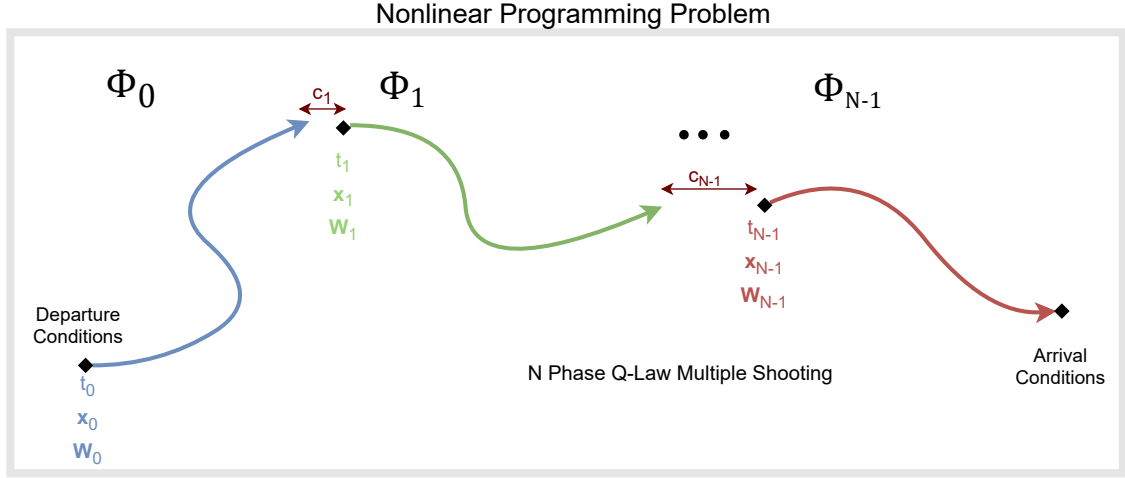


Figure 7.3: Q-Law multiple shooting optimization procedure.

When Q-Law shooting was combined with other trajectory design methods like the Sims-Flanagan model, the NLP decision vector was augmented to include the new design variables, as shown in Equation 7.98.

$$\mathbf{X} = [\mathbf{X}_{SF}; \mathbf{X}_{Q-Law}] \quad (7.98)$$

If true anomaly coasting or a Sims-Flanagan phase was included in the optimization problem, the decision vector was modified to include a variable representing the final mass after the Q-Law propagation, as shown in Equation 7.99. Equations 7.100 and 7.101 show the nonlinear constraint that was enforced to ensure this relationship. The cost function can then be changed to include this variable, directing

the NLP solver to optimize the spacecraft mass after the Q-Law propagation. This was very beneficial because if the cost function listed in Equation 7.82 was used for a mass-optimal problem when a Sims-Flanagan phase was included, it could drastically change the interplanetary solution in an attempt to minimize the mass at the beginning of the spiral, thus yielding the highest specific acceleration and smallest spiral flight time. Conversely, this new cost function let the optimizer vary the interplanetary and spiral decision variables to yield the best possible initial/final mass for the entire trajectory. An example cost function that would be used to maximize final mass for a planetary spiral down problem is shown in Equation 7.102.

$$\mathbf{X} = [a_0, e_0, i_0, \omega_0, \Omega_0, \theta_0, m_0, W_a, W_e, W_i, W_\omega, W_\Omega, t_f, m_1]^T \quad (7.99)$$

$$[a_f, e_f, i_f, \omega_f, \Omega_f, \theta_f, m_f]^T = F(\mathbf{x}_0, \mathbf{W}, t_f) \quad (7.100)$$

$$c_m = m_f - m_1 = 0 \quad (7.101)$$

$$J = -m_1 \quad (7.102)$$

Analytical partial derivatives of the Sims-Flanagan model were derived by Ellison *et al.* and implemented in EMTG [91, 92]. Their work clearly demonstrates the benefits of numerically exact partial derivatives for this model. As a result, accurate sensitivities of the Sims-Flanagan model were obtained through the use of

dual numbers in the JuliaDiff library [54].

7.4 Logistic Function Coasting

Eclipses pose a challenge when designing many-revolution low-thrust trajectories because the thrust must be limited while in the shadow. Conventional Q-Law is a very powerful tool for this problem because event detection root-finding can be applied during the propagation to precisely detect shadow-crossings. Unfortunately, this was not applicable when using analytical derivatives in the Q-Law shooting formulation. The STM was propagated alongside the trajectory, and a root-finding scheme that introduces a discontinuous drop in thrust/power available posed a problem for gradient-based optimizers. As a result, the eclipsing model used within the Q-Law propagation leveraged the same smooth logistic function as the collocation sections of this research [21].

Additionally, Q-Law’s usual coasting mechanism, effectivity, is not analytical and therefore cannot be applied within the shooting problem. However, another logistic function was used to enforce coasting at certain true anomaly ranges. By constraining the thrust or power available to follow Equation 7.103, the mission designer can force the spacecraft into a coast when it enters the specified true anomaly region, as shown in Figure 7.4. In Equation 7.103, θ_L and θ_U represent the lower and upper bounds on the true anomaly coast region, respectively. It was found that $c_\theta = 50$ worked well for larger true anomaly ranges and $c_\theta > 100$ was more effective when the coasting range was less than 10 deg wide.

$$L = \frac{1}{1 + e^{c_\theta(\theta - \theta_L)(\theta_U - \theta)}} \quad (7.103)$$

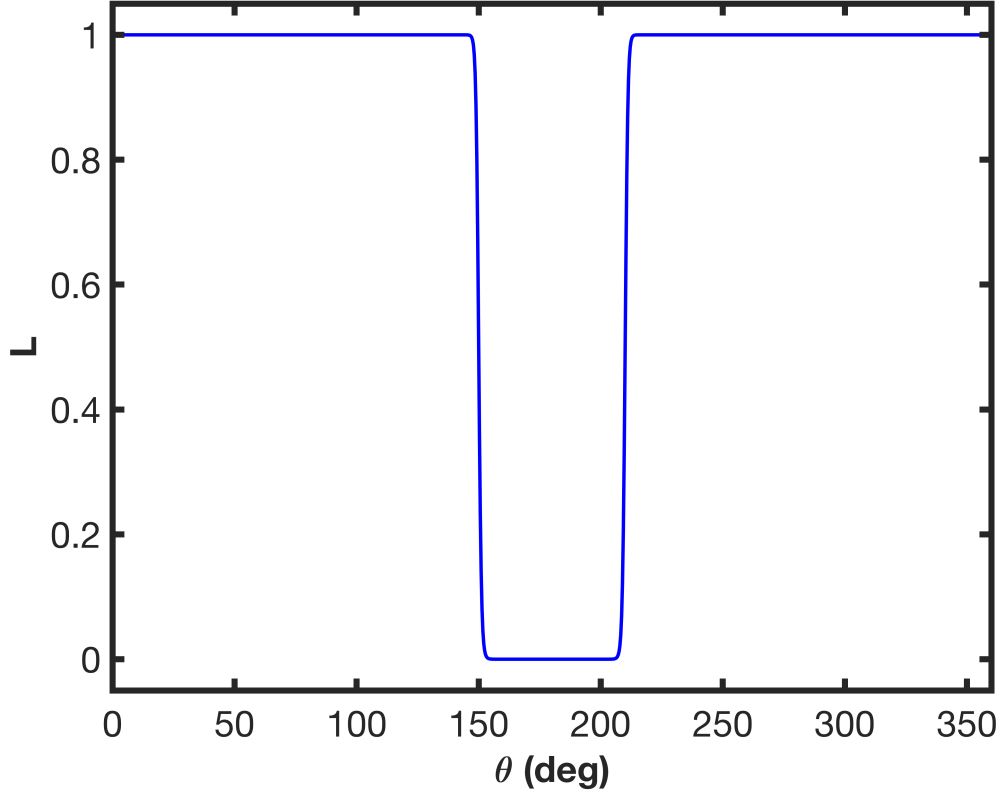


Figure 7.4: Logistic function coasting at $150^\circ \geq \theta \leq 210^\circ$ with $c_\theta = 100$.

A fixed step 4th order Runge-Kutta integration scheme was used to propagate the spacecraft state and STM inside the NLP. This was beneficial as it is faster than higher-order methods, and variable-step integrators require the partial derivatives of the step-size with respect to the current state and are prone to Q-Law thruster chatter. However, the fixed-step integration has errors associated with each step and oversteps both the eclipse shadow entry and exit points and the user-defined coast range. Prior work that employed the logistic eclipse model solved the entire tra-

jectory optimization problem iteratively, placing integration steps closer and closer to the exact shadow crossings. Given Q-Law’s closed-loop nature, the iterative approach was avoided. The fixed step integration errors during the NLP propagation were accepted because the optimal gains/states found by solving the NLP were applied to a high-fidelity (HiFi) model that leveraged variable-step integration and precise event detection. The NLP then served as a means to identify optimal parameters for the lower-fidelity, fixed-step trajectory problem with the intent that they are used later in a high-fidelity model that exactly enforced events like shadow detection and true anomaly coasting and ensured error tolerances were met.

7.5 Example: GTO to GEO Transfer

A well-known GTO to GEO problem was solved to demonstrate the effectiveness of the analytical partial derivatives in the Q-Law shooting formulation [8, 9, 20, 29]. In this problem, the spacecraft was affected by eclipsing and J_2 . The gain combination found by Shannon *et al.* was used as part of the NLP initial guess [29]. The orbits and problem parameters used for this problem are shown in Tables 7.1 and 7.2, respectively.

Table 7.1: Initial and target orbits

Orbit	a (km)	e	incl (deg)	ω (deg)	Ω (deg)
GTO	24363	0.7306	28.5	0	0
GEO	42165	0	0	-	-

The results of this study are summarized in Table 7.3. As expected, formulating the problem using more trajectory phases resulted in an improved cost function

Table 7.2: GTO to GEO transfer dynamics constants and Q-Law parameters.

Constant	Value
Universal Time Departure	March 22, 2000 00:00:00.000 UTC
m_0	1200 kg
I_{sp}	1800 s
P	5 kW
η	0.55
g_0	$9.80665 \times 10^{-3} \text{ km /s}^2$
μ_{\oplus}	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
R_{\oplus}	6378.14 km
R_{\odot}	695500 km
J_2	1086.639×10^{-6}
Q-Law Parameter	Value
Nominal Time Step	1000 secs
W_a (Initial Guess)	2.406
W_e (Initial Guess)	1.786
W_i (Initial Guess)	9.469
W_{ω} (Fixed)	0
W_{Ω} (Fixed)	0
η_a	0
η_r	0
a_{target}	42165 km
e_{target}	0.01
i_{target}	0.01°
a_{tol}	10 km
e_{tol}	0.0025
i_{tol}	0.01°

for both the fixed-step NLP and high-fidelity solutions, with the exception of the 3-phase high-fidelity case, which saw a small increase in TOF. When the optimized gains were used in the high-fidelity model, the resulting flight times were generally very similar to the NLP results. Additionally, the STM approach has clear benefits over finite differencing. In each case, the analytical STM resulted in satisfied optimality conditions whereas the finite differencing cases required more iterations and in some cases, could not converge to an optimal solution. Figure 7.5 shows two GTO-GEO trajectories. The left trajectory is the NLP solution found using

Table 7.3: Minimum-time GTO-GEO problem using NLP wrapped Q-Law.

Problem Setup	NLP Cost (Days)	HiFi TOF (Days)	# of Iterations	Optimality Conditions
Initial Guess	119.73	119.64	N/A	N/A
1 Phase STM	119.55	119.59	10	Satisfied
1 Phase FD	119.55	119.59	13	Satisfied
2 Phases STM	118.93	118.91	56	Satisfied
2 Phases FD	118.93	118.91	80	Not Satisfied
3 Phases STM	118.74	119.08	105	Satisfied
3 Phases FD	118.74	119.08	125	Satisfied
4 Phases STM	118.61	118.76	171	Satisfied
4 Phases FD	N/A	N/A	N/A	N/A*
Graham & Rao[20]	N/A	121.22	N/A	N/A
100% Collocation[29]	N/A	118.62	N/A	N/A
Orbital Averaging[8, 9]	N/A	118.35	N/A	N/A

* No solutions found for the 4 phase finite differencing case.

a fixed-step integrator with low-fidelity eclipse detection. The right trajectory was produced by applying the NLP determined gains to the high-fidelity model, ensuring that integration tolerances were met and eclipse crossings were detected exactly.

The cost function approaches the known orbital averaging solution and agrees well with the optimal collocation solution found by Shannon *et al.* [29]. It should be noted that these Q-Law solutions were subjected to a convergence tolerance and targeted slightly eccentric and inclined orbit to avoid singularities in Gauss’s equations. The GEO boundary constraints were not exactly satisfied as they were in the reference solutions computed using orbital averaging and collocation.

7.6 Example: Lunar Transfer

Rideshares to GTO can provide a cost efficient starting point for a spacecraft to transfer to the Moon, as demonstrated by the SMART-1 mission. Shannon *et al.* introduced the use of backward propagated Q-Law to rapidly design these spiral transfers to the Moon, as described in Chapter 5 [58, 59]. One challenge with this method was targeting a specific mass and epoch after the backward propagation. A MOEA was shown to effectively reduce these errors enough for preliminary design studies, but in general it cannot eliminate them all together. To show how this problem can be addressed with Q-Law shooting, the backward Q-Law phase of the Lunar trajectory design process was isolated and backward Q-Law shooting was used to target a specific mass and epoch to within NLP tolerance.

In this problem, the spacecraft departed from a GTO state with a fixed true

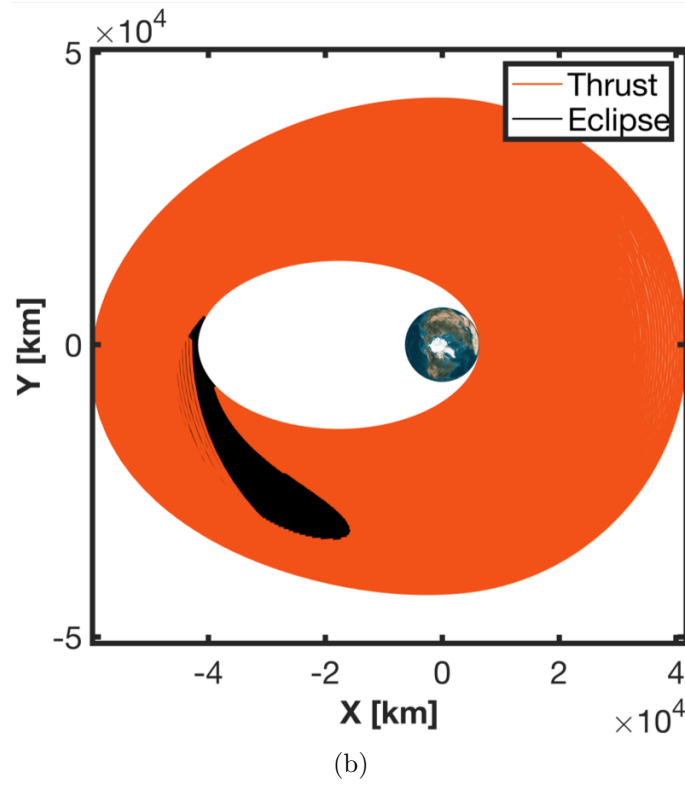
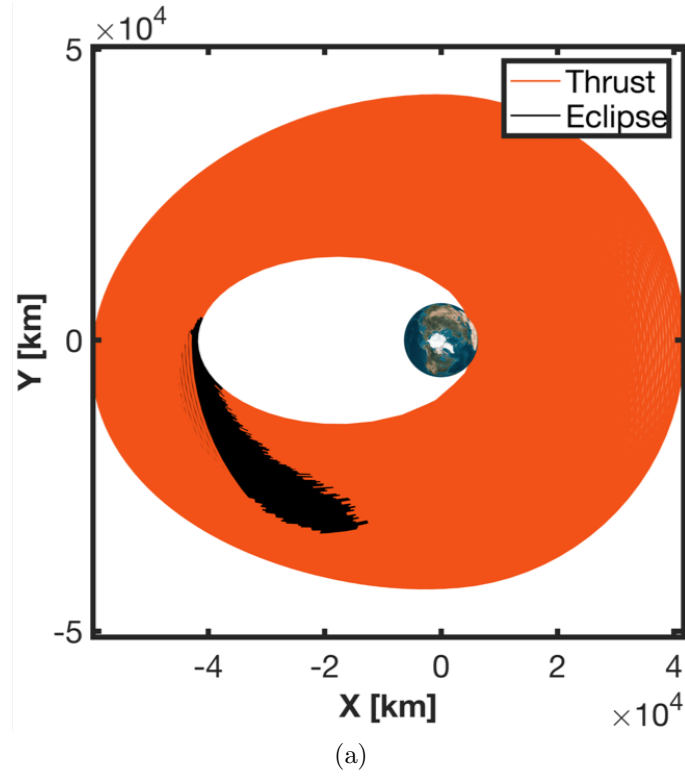


Figure 7.5: GTO-GEO Trajectory solutions. The NLP fixed-step integrator solution with low-fidelity eclipsing is shown on the top. The high-fidelity solution is shown on the bottom.

anomaly, mass, and epoch. A velocity-vector thrusting spiral was used to exit the radiation belts as quickly as possible and reduce the problem size. Backward Q-Law was run from a patch point near the Moon to the velocity-vector thrusting spiral terminal state, and forward Q-Law was run from the Lunar patch point down to the target orbit. Using the evolutionary algorithm results from Shannon *et al.* for a specific departure scenario, the backward Q-law shooting targeted the velocity-vector thrusting spiral terminal state, mass, and epoch, as shown in Figure 7.6. Additional constraints were added to ensure the mass and epoch after the backward Q-Law propagation match the desired values. In this work, only one specific launch scenario out of the sweep performed by Shannon *et al.* was considered. The initial guess for the Q-Law gains came from the MOEA results for this launch scenario. The Earth departure and Lunar target orbits are listed in Table 7.4, and the specific problem parameters used are listed in Table 7.5. The resulting Lunar transfer trajectory was shown in Figure 7.7.

This solution has a total flight time of 144.1 days and delivers 141.04 kg of spacecraft mass to the Lunar target orbit. Using Q-Law within an NLP can successfully satisfy terminal mass and epoch constraints to within NLP tolerance. This new approach introduced the capability to solve Q-Law trajectories with fixed final masses and epochs.

Table 7.4: Initial and target orbits

Orbit	a (km)	e	incl (deg)	ω (deg)	Ω (deg)	θ (deg)
Earth Departure	24363	0.7306	27	0	0	0
Lunar Target	7238.0	0.621	90	270	-	-

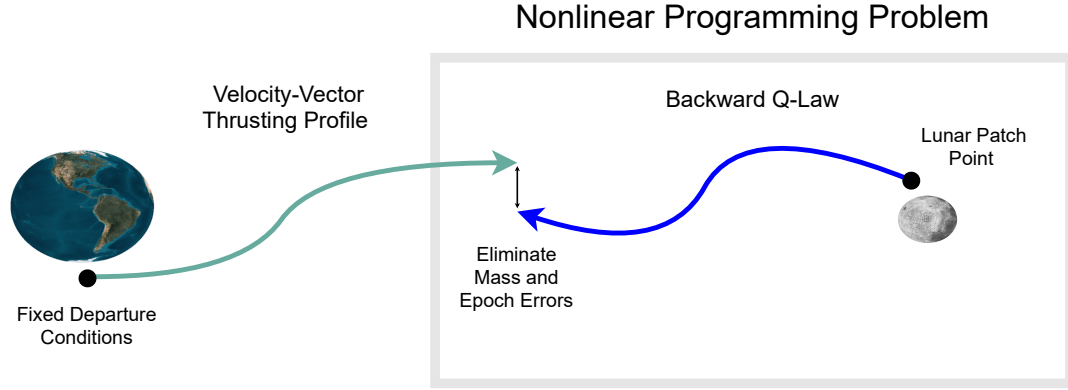


Figure 7.6: Backward Q-Law shooting problem setup to target velocity-vector spiral terminal state.

Table 7.5: Low-thrust Lunar transfer dynamics constants and Q-Law parameters.

Constant	Value
Universal Time Departure	May 31, 2025 00:00:00.000 UTC
m_0	180 kg
I_{sp}	1760 s
Thrust	0.6 N
Duty Cycle	95 %
g_0	$9.80665 \times 10^{-3} \text{ km/s}^2$
μ_{\oplus}	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
μ_{\odot}	$1.32712440018 \times 10^{11} \text{ km}^3/\text{s}^2$
μ_{ζ}	$4.90486 \times 10^3 \text{ km}^3/\text{s}^2$
R_{\oplus}	6378.14 km
R_{\odot}	695500 km
R_{ζ}	1737.5 km
J_2	1086.639×10^{-6}
J_3	-2.565×10^{-6}
J_4	-1.608×10^{-6}
Q-Law Parameter	Value
Nominal Time Step	1000 secs
# of Shooting Phases	1
W_a (Initial Guess)	7.668
W_e (Initial Guess)	7.529
W_i (Initial Guess)	1.059
W_{ω} (Initial Guess)	0.8464
W_{Ω} (Initial Guess)	2.426
η_a	0
η_r	0
a_{target}	93629.37 km
e_{target}	0.38
i_{target}	26.96°
ω_{target}	15.10°
Ω_{target}	353.81°
θ_{target}	170.60°
a_{tol}	10 km
e_{tol}	0.01
i_{tol}	0.1°
ω_{tol}	0.1°
Ω_{tol}	0.1°
θ_{tol}	0.1°

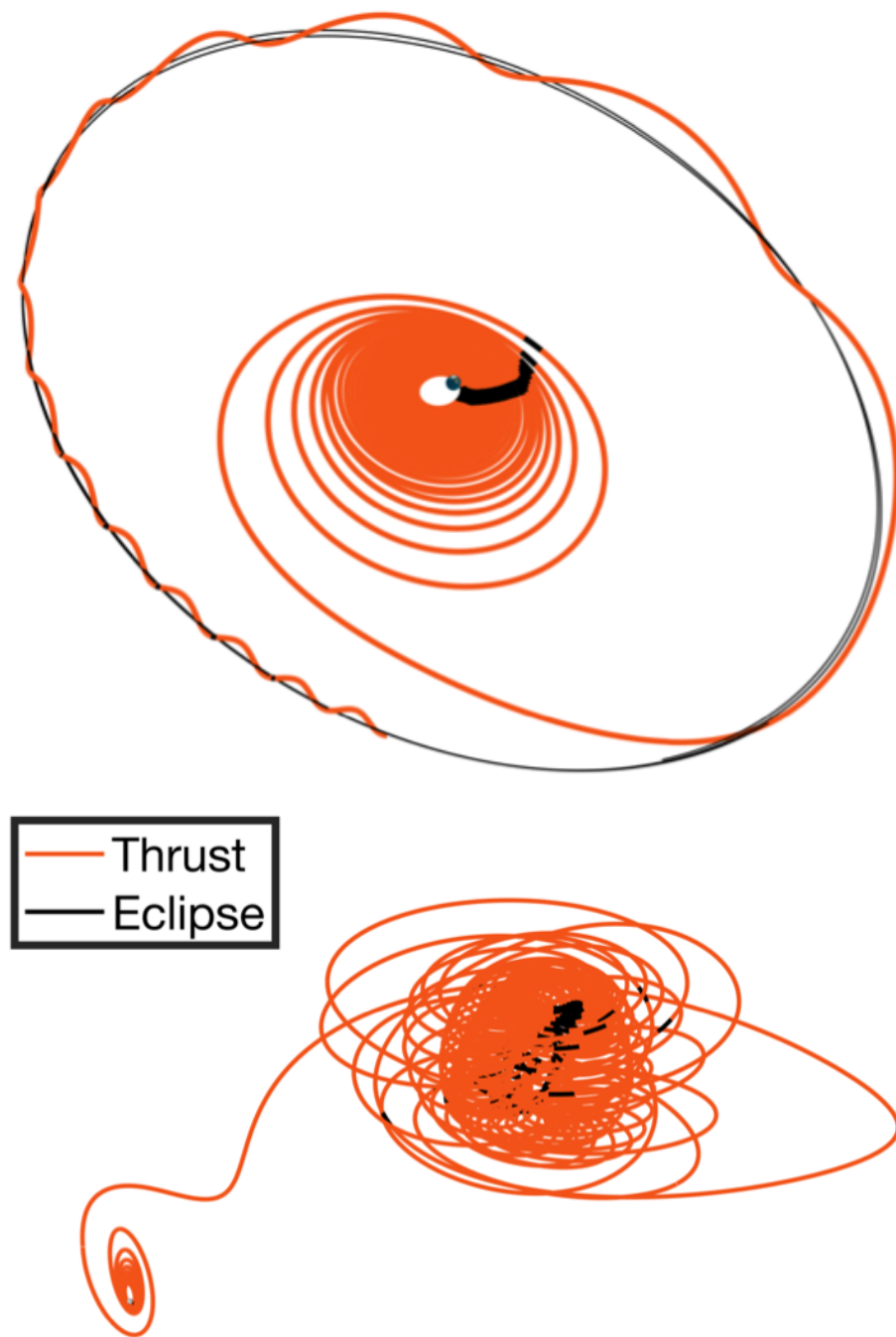


Figure 7.7: Low-thrust transfer to the Moon in the Earth-centered inertial frame (top) and the selenocentric rotating frame (bottom).

7.7 Example: Direct Launch and Capture at Mars

The Q-Law shooting formulation can be combined with interplanetary design techniques like the Sims-Flanagan model. Doing so includes the spiral design in the same NLP as the interplanetary design, resulting in an end-to-end optimization of the entire trajectory. This problem is solved as a single NLP with two phases: the interplanetary phase and the spiral phase. This was demonstrated on a Mars transfer and capture trajectory. In this problem, a SmallSat spacecraft was launched directly from Earth and used a low-thrust engine to rendezvous and spiral down at Mars, as depicted in Figure 7.8. This problem represented the scenario where a SmallSat rideshares with a larger interplanetary launch to Mars. Nonlinear constraints were placed on the match point state to ensure that the initial spiral state was loosely captured and near the planet’s SOI. The specific problem parameters used to solve this problem are listed in Table 7.6.

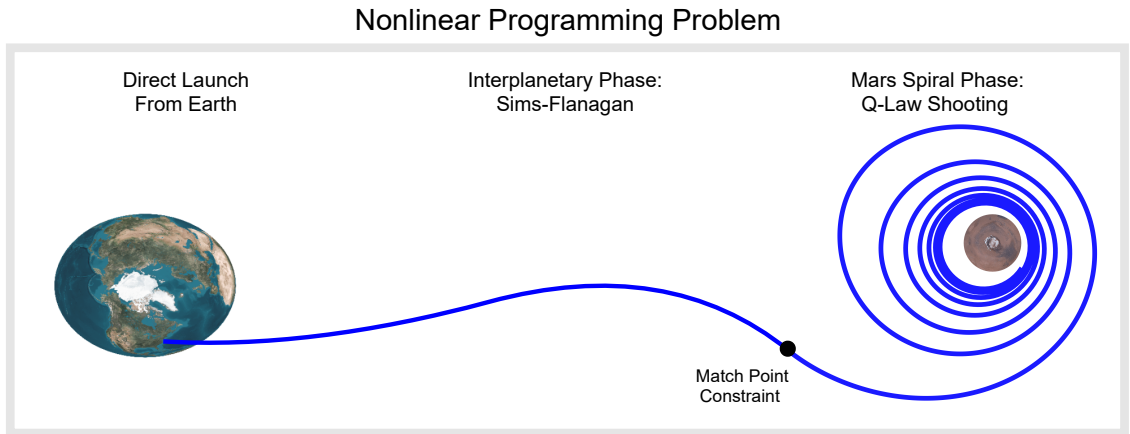


Figure 7.8: Sims-Flanagan + Q-Law Shooting problem setup for a Mars transfer and capture.

This problem was solved both with and without true anomaly coasting using

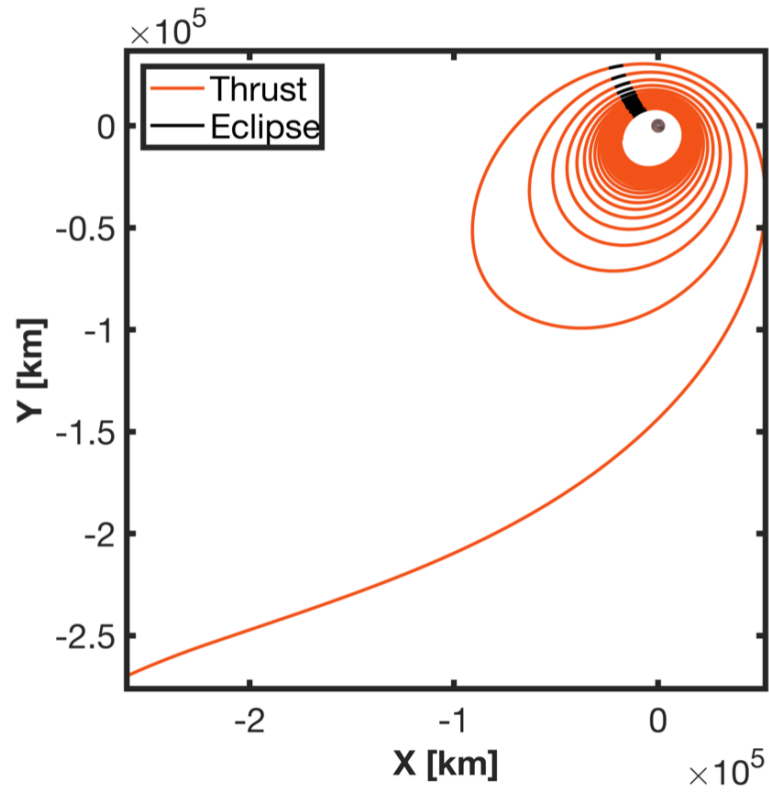
Table 7.6: Mars transfer and capture dynamics constants and problem parameters.

Constant	Value
g_0	$9.80665 \times 10^{-3} \text{ km/s}^2$
R_{Mars}	3389.0 km
μ_{Mars}	$42828.37 \text{ km}^3/\text{s}^2$
R_{\odot}	695500 km
μ_{\odot}	$1.32712440018 \times 10^{11} \text{ km}^3/\text{s}^2$
Mission Parameter	Value
Launch Window Open	July 1, 2020 00:00:00.000 UTC
m_0	200 kg
I_{sp}	1760 s
Thrust	0.60 N
Q-Law Parameter	Value
Nominal Time Step	1000 secs
# of Shooting Phases	1
W_a (Initial Guess)	1.0
W_e (Initial Guess)	3.0
W_i (Initial Guess)	1.0
W_{ω} (Fixed)	0
W_{Ω} (Fixed)	0
a_{target}	16353 km
e_{target}	0.4354
i_{target}	27.5°
a_{tol}	50 km
e_{tol}	0.05
i_{tol}	0.1°

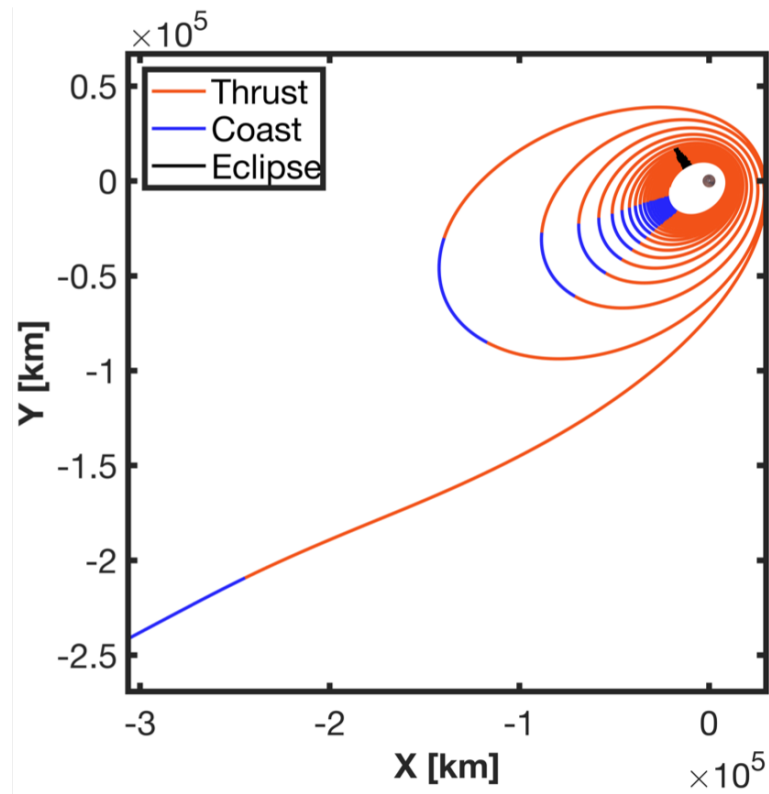
the logistic function. When coast arcs were included, they were centered around apoapsis as this is typically where coasting is most optimal when changes in semi-major axis are required. The resulting trajectories are shown in Figure 7.9 with the results summarized in Table 7.7. It can be seen that true anomaly coasting is effective at increasing the final mass delivered to the Mars target orbit.

Table 7.7: Mars transfer and capture results. TOFs refer to the spiral phase only.

Case	NLP m_f (kg)	NLP TOF (days)	HiFi m_f (kg)	HiFi TOF (days)
No Coasting	159.35	55.19	159.34	55.22
$170^\circ \geq \theta \leq 190^\circ$ Coasting	161.65	55.65	161.71	55.96
$160^\circ \geq \theta \leq 200^\circ$ Coasting	163.60	62.28	163.67	64.01

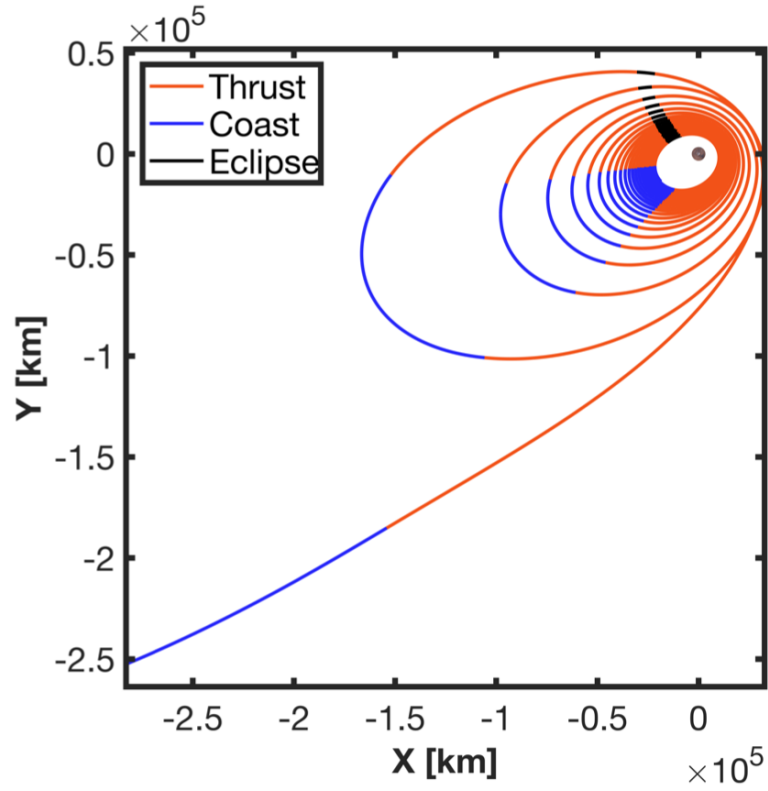


(a) Mars capture spiral with no coasting.

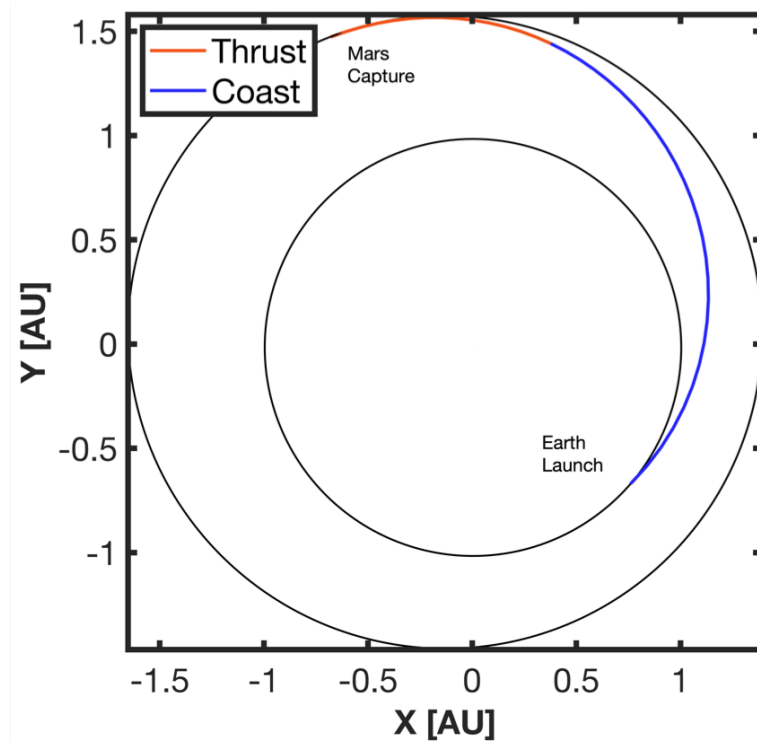


(b) Mars capture spiral with coasting between $170^\circ \geq \theta \leq 190^\circ$.

Figure 7.9: Low-thrust Martian capture trajectories.



(c) Mars capture spiral with coasting between $160^\circ \geq \theta \geq 200^\circ$.



(d) Interplanetary trajectory to Mars. Launch $v_\infty = 4.23$ km/s.

Figure 7.9: Low-thrust Martian capture trajectories.

7.8 Example: Lunar Swingby Escape to Comet 45P

The last example problem focused on a low-thrust Lunar swingby escape problem for a SmallSat that departs from GTO and performs a flyby of Comet 45P. This class of trajectory has been shown to be very enabling for SmallSats by offering access to interplanetary space without reliance on a rideshare with an interplanetary launch [72]. In this problem, backward propagated Q-Law was used to generate the spiral from the Lunar swingby to GTO, and an additional phase for the Earth-centered keplerian propagation to Earth's SOI was added between the Lunar swingby and the interplanetary phase. This trajectory type is illustrated in Figure 7.10. The initial spiral state was constrained such that it must obey the patched-conic ZSOI gravity-assist model with the outgoing state, as described in Equations 6.6 to 6.8. The specific parameters used to solve this problem are listed in Table 7.8.

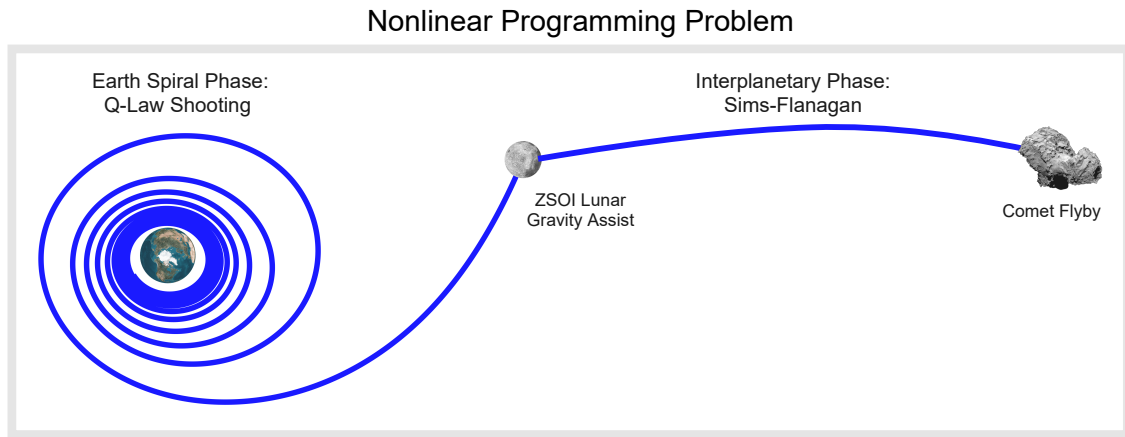


Figure 7.10: Sims-Flanagan + Q-Law Shooting problem setup for a Lunar swingby spiral escape. Comet Image: ESA/Rosetta/NAVCAM, CC BY-SA IGO 3.0

In this problem, the cost function was defined to be the initial spacecraft mass at GTO, therefore minimizing the propellant required to deliver the target mass to

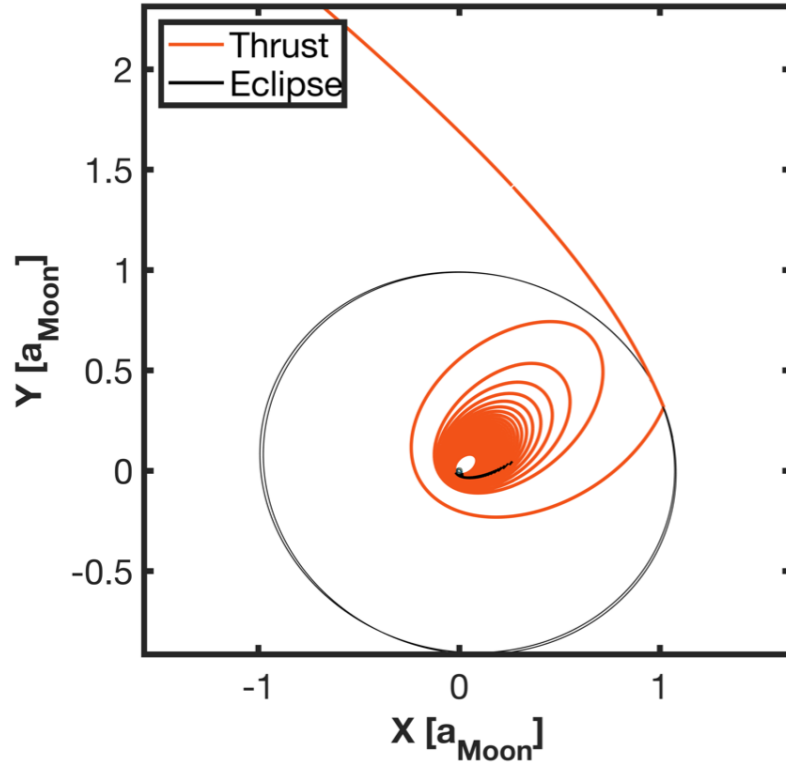
Table 7.8: Lunar swingby escape dynamics constants and problem parameters.

Constant	Value
g_0	$9.80665 \times 10^{-3} \text{ km/s}^2$
μ_{\oplus}	$3.9860047 \times 10^5 \text{ km}^3/\text{s}^2$
μ_{\odot}	$1.32712440018 \times 10^{11} \text{ km}^3/\text{s}^2$
$\mu_{\mathcal{L}}$	$4.90486 \times 10^3 \text{ km}^3/\text{s}^2$
R_{\oplus}	6378.14 km
R_{\odot}	695500 km
$R_{\mathcal{L}}$	1737.5 km
J_2	1086.639×10^{-6}
Mission Parameter	Value
Launch Window Open	Nov 1, 2029 00:00:00.000 UTC
Flyby Epoch Bounds	Aug 1 2032 - Jan 31 2033
Max Flyby Velocity	6 km/s
m_f	100 kg
I_{sp}	1760 s
Thrust	0.60 N
Minimum Swingby Altitude	200 km
Q-Law Parameter	Value
Nominal Time Step	1000 secs
# of Shooting Phases	1
W_a (Initial Guess)	1.0
W_e (Initial Guess)	5.0
W_i (Initial Guess)	1.0
W_{ω} (Fixed)	0
W_{Ω} (Fixed)	0
a_{target}	24363 km
e_{target}	0.7306
i_{target}	28.5°
a_{tol}	50 km
e_{tol}	0.05
i_{tol}	0.1°

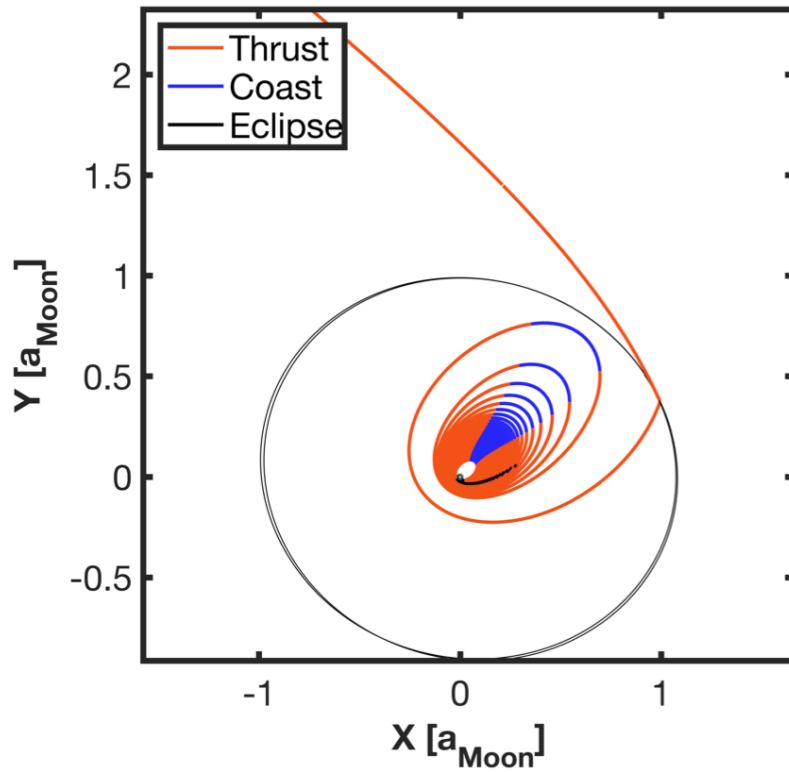
the comet flyby. Figure 7.11 shows the resulting trajectories, and Table 7.9 summarizes the results of the coasting and non-coasting solutions. Significant propellant savings were observed when true anomaly coasting is applied near apoapsis.

Table 7.9: Lunar gravity-assist and comet flyby trajectory results. TOFs refer to the spiral phase only.

Case	NLP m_0 (kg)	NLP TOF (days)	HiFi m_0 (kg)	HiFi TOF (days)
No Coasting	187.56	137.61	187.42	137.08
$170^\circ \geq \theta \leq 190^\circ$ Coasting	179.34	139.93	178.87	139.57
$160^\circ \geq \theta \leq 200^\circ$ Coasting	172.87	154.93	172.62	155.27

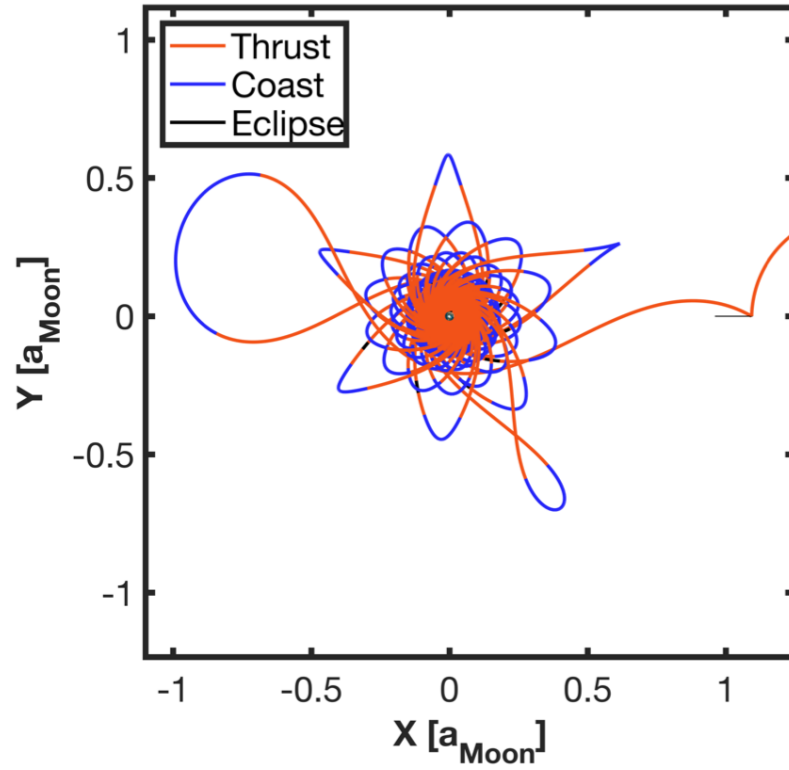


(a) Spiral Lunar flyby trajectory with no coasting.

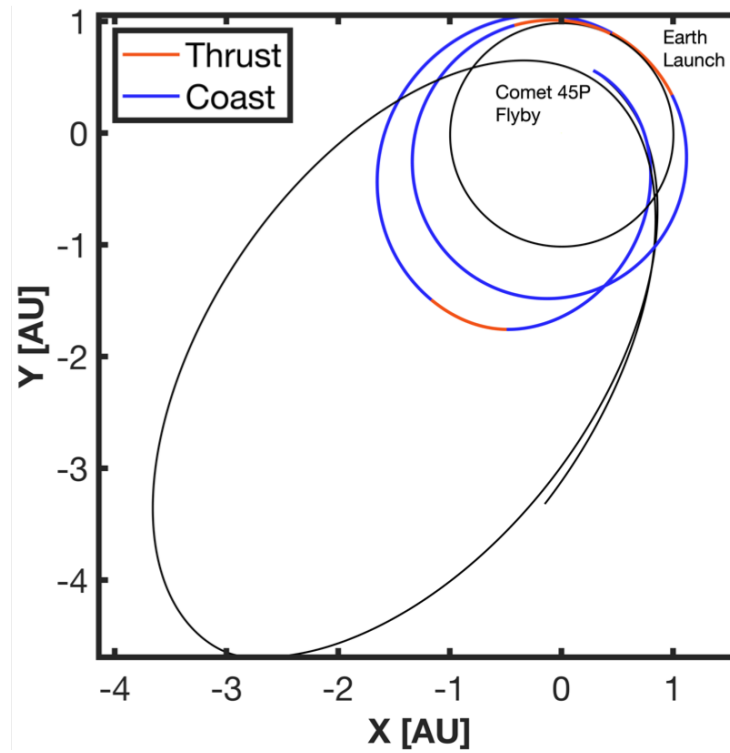


(b) Spiral Lunar flyby trajectory with coasting between $170^\circ \geq \theta \leq 190^\circ$.

Figure 7.11: Low-thrust escape trajectories to Comet 45P.



(c) Spiral Lunar flyby trajectory in the Earth-Moon rotating frame with coasting between $160^\circ \geq \theta \geq 200^\circ$.



(d) Interplanetary trajectory to Comet 45P.

Figure 7.11: Low-thrust escape trajectories to Comet 45P.

Chapter 8: Conclusions

Rideshare spacecraft missions to cislunar and interplanetary space offer enabling trajectories for small-budget exploration efforts. This research extended the capabilities and application of the existing low-thrust guidance algorithm Q-Law within the many-revolution transfer problem to address these mission types. Rapid trade study tools as well as high-fidelity hybrid optimization methods were developed to provide a useful framework to design these complex low-thrust trajectories.

8.1 Summary

First, Q-Law was used to rapidly produce a transfer trajectory from GTO to GEO to demonstrate that the Q-Law solution provides a strong initial guess for direct collocation. The location of the patch point was varied along the Q-Law solution to demonstrate the existence of a trade space between computation time and solution optimality as the patch point is moved. The Q-Law gains were optimized using a genetic algorithm, and the trajectory optimization successfully produced optimal single phase trajectories from GTO-GEO. When the patch point was placed at the start of the Q-Law trajectory, the resulting solution exceeded the results of Graham and Rao [20] and closely approached the known orbital averaging

solution. However, when eclipses are included in the optimization phase, the solution becomes more sensitive and computation time increases. Using the high-fidelity Q-Law to generate a minimum-time solution is very fast and relatively close to the global optimum. Therefore, it may be desirable to use Q-Law as a means to detect eclipses, and the remaining, non-eclipsed trajectory phase can quickly be optimized using the Q-Law solution as the initial guess. The results of this study demonstrate that solutions that deviate less than 1% from the orbital averaged time-optimal solution can be produced with this method without significant computation time. This method was also used to produce a mass-optimal solution with the Q-Law TOF used as an endpoint constraint in the GPOPS-II mass-optimal problem. The mass-optimal solution demonstrated bang-bang control and minor propellant savings over the time-optimal case. This hybrid approach provides a computationally efficient means to produce near-optimal many revolution trajectories.

Low-thrust spiral transfers to the Moon represent a particularly difficult yet highly desirable example of the many-revolution problem. In addition to the large number of revolutions and possible eclipsing, proper phasing with the Moon is needed for successful capture. However, this trajectory type can be beneficial for smaller spacecraft looking for discounted launch opportunities through a rideshare to common Earth parking orbits. This work introduced a novel application of Q-Law that rapidly generates Earth-Moon spiral trajectories by combining forward and backward Q-Law propagation. In this approach, a patch point was selected in Lunar orbital element space and used as the transition between the Earth and Moon spiral phases. The Earth-centered phase started at the patch point and propagates

backwards, solving for a trajectory from the departure orbit to the patch point. In the Moon-centered phase, forward propagated Q-Law solved for a trajectory from the patch point to the target final orbit, resulting in a continuous trajectory from the Earth to the Moon.

High-fidelity Q-Law solutions require minimal computation time to generate and provide excellent initial guesses for direct collocation, representing a significant improvement in Lunar spiral initial guess generation techniques. This was demonstrated by solving a trajectory design problem inspired by the SMART-1 mission. Using this method, a time-optimal solution was found that is four days shorter than the solution previously identified by Betts, and a mass-optimal solution was found that exhibits bang-bang control throughout the entire trajectory, resulting in more than 15 kg of propellant savings over Betts’s mass-optimal result.

The efficiency of the forward-backward Q-Law approach also makes it an effective trade study tool, allowing mission designers to sweep through the trajectory trade space much faster than previously possible. A new trajectory design problem was created with modern epochs and an ESPA-class spacecraft. To explore the design space for this mission scenario, forward-backward Q-Law was wrapped in a multi-objective evolutionary algorithm to minimize mass and epoch errors and produce close to time-optimal solutions. Different departure epochs and right ascensions were explored to determine the effect on flight time and number of eclipses. Q-Law’s effectivity coasting was then used on both the forward and backward Q-Law phases to produce a propellant usage Pareto front for a specific departure case. Using these solutions as an initial guess, optimized results were compared to the Q-Law effec-

tivity solutions and were found to be within a few kg of each other. Additionally, an all-backward Q-Law solver wrapped in a multi-objective evolutionary algorithm was demonstrated for trajectories from LEO to LLO. When arrival epoch and delivered mass are fixed, backward Q-Law can start from the target Lunar orbit and spiral out to a transition point where Earth-centered backward Q-Law begins and spirals down to the departure orbit. In this approach, departure mass and epoch can vary, allowing the multi-objective evolutionary algorithm to directly identify Pareto optimal transfers. Low-thrust LEO-LLO transfers contain extremely high numbers of revolutions and eclipses that could over-burden other design techniques. The approach developed in this research overcomes these issues and efficiently explores the trajectory trade space. As previously noted, many-revolution trajectory optimization techniques can be highly sensitive and require favorable initial guesses, which may be difficult to construct for complicated trajectories. Conversely, rapid exploration of the problem space using Q-Law can produce solutions that, when paired with direct optimization, produce near-optimal solutions in minimal time regardless of eclipsing, prior solution knowledge, or problem complexity.

Additionally, this work demonstrated an approach for designing spiral escape trajectories that leverage Lunar gravity assists. Single and double flybys were considered, with the flyby sequence and interplanetary leg of the mission designed using the perturbed Sims-Flanagan low-thrust model and Monotonic Basin Hopping implemented in EMTG. Low-thrust spirals from GTO were connected to the rest of the trajectory through the first Lunar gravity assist. These spirals were designed using backwards propagated Q-Law starting at the Moon. This approach was demon-

strated by designing single and double lunar gravity assist trajectories as well as conventional spiral escape trajectories for a sample SmallSat mission to Comet 45P. As expected, the double flyby cases achieved the highest escape C_3 , making the interplanetary phases more efficient. Both the single and double swingby scenarios required significantly less propellant than the conventional spiral escape case, offering efficient escape trajectory options for interplanetary SmallSats.

Lastly, partial derivatives of the Q-Law thrust vector calculation with respect to the control law input gains and the spacecraft state were derived. These derivatives were used to generate a trajectory STM, which provided the optimizer with the exact sensitivities of the terminal Q-Law state to the decision vector. Using this STM, a Q-Law shooting formulation was developed and applied to various low-thrust transfer problems. Logistic functions were used to approximate eclipses and enforce coast arcs at specific true anomaly ranges.

A well-known GTO-GEO transfer was solved using a variable number of shooting phases as well as finite differencing for comparison. The results from this problem clearly demonstrated the benefits of providing numerically exact partial derivatives to the NLP solver and showed that gradient-based gain tuning can be very effective when a good initial guess is available. As more phases were introduced, the GTO-GEO NLP cost function approached the known orbital averaging solution. Also, backward Q-Law shooting was used to help design low-thrust transfers to the Moon. The shooting algorithm was used in a backward Q-Law phase to target the specific mass and epoch of the terminal state of a velocity vector thrusting spiral. In prior studies, targeting these quantities proved challenging and evolutionary al-

gorithms had to be employed to reduce possible errors. However, Q-Law shooting is capable of directly varying the starting epoch and mass to enforce boundary constraints to within NLP tolerance. This added capability can be used to solve Q-Law trajectories with fixed final masses and epochs. Additionally, the Q-Law trajectory sensitivities were used to combine the shooting method with the well-known Sims-Flanagan interplanetary model, allowing for end-to-end trajectory optimization in one NLP. This work made this technique viable for spiral escape/capture trajectories. It provided more accurate flight time and propellant requirement estimates as well as the ability to target eccentric orbits for this kind of design. This method was applied to a Mars transfer and spiral down trajectory and a Lunar swingby escape trajectory to Comet 45P. In both scenarios, true anomaly coasting proved to be effective at reducing propellant requirements. In all cases, the optimizer results were refined with a high-fidelity propagator with precise eclipse/true anomaly detection to demonstrate the utility of the NLP generated trajectories as an input to a high-fidelity model. In general, the NLP solutions compared closely to the high-fidelity results.

Spiral transfers to the Moon, either for capture or gravity assist present an enabling trajectory option for the SmallSat community, allowing for reduced launch cost and launch window flexibility. The methods described here provide mission designers with the ability to rapidly generate spiral transfers in the Earth-Moon system with minimal effort and computation time. This produces better initial guesses for trajectory optimization and more efficient trade studies during the preliminary mission design phase. Q-Law proves to be an effective tool for Earth-Moon

spiral design and offers critical mission design capabilities that will be needed as these mission types becomes more common. Furthermore, the partial derivatives presented in this paper prove to be effective when applied to the Q-Law shooting problem. These derivatives were used to develop a new hybrid trajectory design technique. By combining direct optimization, Lyapunov control, and interplanetary design techniques, this research improves many-revolution spiral design by offering gradient-based gain tuning and end-to-end trajectory optimization capabilities.

8.2 Publications and Presentations

Journal Publications

- Shannon, J., Ellison, D., and Hartzell, C., “Analytical Partial Derivatives of the Q-Law Guidance Algorithm”, Journal of Astronautical Sciences, 2021. (Submitted)
- Shannon, J., Ozimek, M., Atchison, J., and Hartzell, C, “Rapid Design and Exploration of High-Fidelity Low-Thrust Transfers to the Moon”, Journal of Spacecraft and Rockets, 2021. (Submitted)
- Shannon, J., Atchison, J., Villac, B., Rogers, G., and Ozimek, M., “Mission Design for the 2020 Mercury Lander Decadal Survey”, Journal of Astronautical Sciences, 2020. (Accepted)
- Shannon, J., Ozimek, M., Atchison, J., and Hartzell, C, “Q-Law Aided Direct Trajectory Optimization of Many-Revolution Low-Thrust Transfers, ” Journal of Spacecraft and Rockets, Vol. 57, No. 4 (2020), pp. 672-682 doi: doi/abs/10.2514/1.A34586

Conference Presentations

- Shannon, J., Ellison, D., and Hartzell, C., “Analytical Partial Derivatives of the Q-Law Guidance Algorithm,” 2021 AAS/AIAA Space Flight Mechanics Conference, AAS Paper 21-274, 2021.
- Shannon, J., Ellison, D., and Hartzell, C., “Exploration of Low-Thrust Lunar Swingby Escape Trajectories,” 2021 AAS/AIAA Space Flight Mechanics Conference, AAS Paper 21-273, 2021.
- Shannon, J., Atchison, J., Villac, B., Rogers, G., and Ozimek, M., “Mission Design for the 2020 Mercury Lander Decadal Survey,” 2020 AAS/AIAA Astrodynamics Specialist Conference, 2020.
- Shannon, J., Ozimek, M., Atchison, J., and Hartzell, C., “Rapid Design and Exploration of High-Fidelity Low-Thrust Transfers to the Moon,” 2020 IEEE Aerospace Conference, IEEE, 2020, pp. 1–11.
- Shannon, J., Ozimek, M., Atchison, J., and Hartzell, C., “Q-Law Aided Direct Trajectory Optimization For The High-Fidelity, Many-Revolution Low-Thrust Orbit Transfer Problem,” 2019 AAS/AIAA Space Flight Mechanics Conference, AAS Paper 19-448, 2019.

8.3 Future Work

Several avenues for further investigation immediately present themselves. Q-Law single shooting was demonstrated on several complex low-thrust transfer problems, including an Earth-Moon spiral, single Lunar gravity assist, and a Mars cap-

ture spiral. Further analysis should examine the effects on these transfers when additional shooting phases are introduced. A cost function improvement should be expected, but additional phases can increase the NLP complexity/sensitivity and run time.

Furthermore, the application of Q-Law shooting to Lunar transfers should be extended to include the patch point location and the forward propagation phase. In this work, Q-Law shooting was only used for the backward phase as a demonstration of full state, mass, and endpoint targeting capabilities. Initial efforts to include a variable patch point and the forward propagated phase in the NLP were unsuccessful. The variations in the patch point location and gains combined with n-body effects from the Earth and Sun cause the spacecraft to escape the Moon, prohibiting convergence at the target Lunar orbit. Future efforts could investigate whether modified equinoctial elements provide a more stable state representation for this problem.

The Sundman transformation proved to be an important inclusion for the collocation efforts in this research. It effectively reduces the dynamic sensitivity of the many-revolution problem and better distributes the mesh points. A logical next step would be to investigate the benefits of applying the Sundman transformation to Q-Law shooting. Several transfers investigated in this work have periods of high eccentricity, and a Sundman transformed independent variable would likely improve solution accuracy and require fewer integration steps, therefore reducing computation time for a given trajectory evaluation. However, this could present problems if the spacecraft is in a highly nonlinear region of the Earth-Moon three-body problem

or near escape from its central body, as a reasonable step size in an anomaly variable can result in an extremely large step in time. If some prior knowledge of the solution type is known, the multiple shooting formulation could be well-equipped to overcome this by varying the independent variable between shooting segments. In this approach, the propagation can proceed in time in the segment(s) containing the sensitive region, and the remaining segments can leverage Sundman transformed dynamics. The analytical derivatives of the Sundman transformed dynamics will need to be derived to generate an STM for the constraint sensitivities.

Bibliography

- [1] Giuseppe D Racca, Bernard H Foing, and Marcello Coradini. Smart-1: The first time of europe to the moon. In *Earth-Moon Relationships*, pages 379–390. Springer, 2001.
- [2] Johan Schoenmaekers. Post-launch optimisation of the smart-1 low-thrust trajectory to the moon. In *18th International Symposium on Space Flight Dynamics*, volume 548, page 505, 2004.
- [3] Bethany L Ehlmann, Rachel L Klima, Diana L Blaney, Neil E Bowles, Simon B Calcutt, James L Dickson, K Donaldson Hanna, Christopher S Edwards, Rory Evans, Robert O Green, et al. Lunar trailblazer: A pioneering smallsat for lunar water and lunar geology. *AGUFM*, 2019:A43F–05, 2019.
- [4] JS Goodwin and Peter Wegner. Evolved expendable launch vehicle secondary payload adapter. In *Proceeding of the AIAA/USU Conference on Small Satellites. SSC01-X-6. Albuquerque, USA*, 2001.
- [5] Natasha Bosanac, Andrew D Cox, Kathleen C Howell, and David C Folta. Trajectory design for a cislunar cubesat leveraging dynamical systems techniques: The lunar icecube mission. *Acta Astronautica*, 144:283–296, 2018.
- [6] Jean Albert Kechichian. Trajectory optimization using nonsingular orbital elements and true longitude. *Journal of Guidance, Control, and Dynamics*, 20(5):1003–1009, 1997.
- [7] Andrew S. Feistel and Christopher L. Ranieri. Modeling perturbations and operational considerations when using indirect optimization with equinoctial elements. In *AAS/AIAA Space Flight Mechanics Meeting*, pages 1737–1756, 2009.
- [8] Craig A Kluever and BA Conway. Low-thrust trajectory optimization using orbital averaging and control parameterization. *Spacecraft Trajectory Optimization*, pages 112–138, 2010.

- [9] Robert Falck and John Dankanich. Optimization of low-thrust spiral trajectories by collocation. In *AIAA/AAS Astrodynamics Specialist Conference*, page 4423, 2012.
- [10] Jonathan D Aziz, Jeffrey S Parker, Daniel J Scheeres, and Jacob A Englander. Low-thrust many-revolution trajectory optimization via differential dynamic programming and a sundman transformation. *The Journal of the Astronautical Sciences*, 65(2):205–228, 2018.
- [11] Richard E Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.
- [12] Wayne A Scheel and Bruce A Conway. Optimization of very-low-thrust, many-revolution spacecraft trajectories. *Journal of Guidance, Control, and Dynamics*, 17(6):1185–1192, 1994.
- [13] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Siam, Philadelphia, PA, 2010.
- [14] Bruce A Conway. *Spacecraft trajectory optimization*, volume 29. Cambridge University Press, 2010.
- [15] John T Betts and Sven O Erb. Optimal low thrust trajectories to the moon. *SIAM Journal on Applied Dynamical Systems*, 2(2):144–170, 2003.
- [16] Sophie Geffroy and Richard Epenoy. Optimal low-thrust transfers with constraints—generalization of averaging techniques. *Acta astronautica*, 41(3):133–149, 1997.
- [17] T Edelbaum, L Sackett, and H Malchow. Optimal low thrust geocentric transfer. In *10th Electric Propulsion Conference*, page 1074, 1973.
- [18] C. Ferrier and R. Epenoy. Optimal control for engines with electro-ionic propulsion under constraint of eclipse. *Acta Astronautica*, 48(4):181–192, 2001.
- [19] John T Betts. Optimal low-thrust orbit transfers with eclipsing. *Optimal Control Applications and Methods*, 36(2):218–240, 2015.
- [20] Kathryn F Graham and Anil V Rao. Minimum-time trajectory optimization of low-thrust earth-orbit transfers with eclipsing. *Journal of Spacecraft and Rockets*, 53(2):289–303, 2016.
- [21] Jonathan Aziz, Daniel Scheeres, Jeffrey Parker, and Jacob Englander. A smoothed eclipse model for solar electric propulsion trajectory optimization. *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, 17(2):181–188, 2019.
- [22] Anastassios Petropoulos. Low-thrust orbit transfers using candidate lyapunov functions with a mechanism for coasting. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pages 748–762, 2004.

- [23] Anastassios E Petropoulos. Refinements to the q-law for the low-thrust orbit transfers. In *AIAA/AAS Space Flight Mechanics Conference. AAS Paper 05-162*, 2005.
- [24] Anastassios Petropoulos and Seungwon Lee. Optimisation of low-thrust orbit transfers using the q-law for the initial guess. In *AAS/AIAA Astrodynamics Specialist Conference and Exhibit*, pages 2229–2248, 2005.
- [25] David A Vallado. *Fundamentals of Astrodynamics and Applications*, volume 12. Springer Science & Business Media, 2001.
- [26] MJH Walker, B Ireland, and Joyce Owens. A set modified equinoctial orbit elements. *Celestial mechanics*, 36(4):409–419, 1985.
- [27] MJH Walker. Erratum - a set of modified equinoctial orbit elements. *Celestial mechanics*, 38(4):391–392, 1986.
- [28] Matthew Berry and Liam Healy. The generalized sundman transformation for propagation of high-eccentricity elliptical orbits. Technical report, Naval Research Lab Washington DC, 2002.
- [29] Jackson L Shannon, Martin T Ozimek, Justin A Atchison, and Christine M Hartzell. Q-law aided direct trajectory optimization of many-revolution low-thrust transfers. *Journal of Spacecraft and Rockets*, 57(4):672–682, 2020.
- [30] Charles H Acton Jr. Ancillary data services of nasa’s navigation and ancillary information facility. *Planetary and Space Science*, 44(1):65–70, 1996.
- [31] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002.
- [32] Gábor I Varga and José M Sánchez Pérez. Many-revolution low-thrust orbit transfer computation using equinoctial q-law including j2 and eclipse effects. *AIAA/AAS Paper*, pages 15–590, 2016.
- [33] Seungwon Lee, Paul Von Allmen, Wolfgang Fink, Anastassios E Petropoulos, and Richard J Terrile. Comparison of multi-objective genetic algorithms in optimizing q-law low-thrust orbit transfers. In *GECCO Conference Late-breaking Paper, Washington, DC*, 2005.
- [34] Seungwon Lee, Paul von Allmen, Wolfgang Fink, AE Petropoulos, and RJ Terrile. Multi-objective evolutionary algorithms for low-thrust orbit transfer optimization. In *Genetic and Evolutionary Computation Conference (GECCO 2005)*, 2005.
- [35] Yang Dalin, Xu Bo, and Gao Youtao. Optimal strategy for low-thrust spiral trajectories using lyapunov-based guidance. *Advances in Space Research*, 56(5):865–878, 2015.

- [36] Da-lin Yang, Bo Xu, and Lei Zhang. Optimal low-thrust spiral trajectories using lyapunov-based guidance. *Acta Astronautica*, 126:275–285, 2016.
- [37] Noble A Hatten. A critical evaluation of modern low-thrust, feedback-driven spacecraft control laws. Master’s thesis, The University of Texas at Austin, 2012.
- [38] DV Lantukh, CL Ranieri, MD DiPrinzio, and PJ Edelman. Enhanced q-law lyapunov control for low-thrust transfer and rendezvous design. In *2017 AAS/AIAA Astrodynamics Specialist Conference*, 2017.
- [39] Richard H Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, revised edition*. American Institute of Aeronautics and Astronautics, 1999.
- [40] Douglas P Hamilton. Motion of dust in a planetary magnetosphere: Orbit-averaged equations for oblateness, electromagnetic, and radiation forces with application to saturn’s e ring. *Icarus*, 101(2):244–264, 1993.
- [41] Michael A Patterson and Anil V Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1, 2014.
- [42] Fengjin Liu, William W Hager, and Anil V Rao. Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction. *Journal of the Franklin Institute*, 352(10):4081–4106, 2015.
- [43] John T Betts. Very low-thrust trajectory optimization using a direct sqp method. *Journal of Computational and Applied Mathematics*, 120(1-2):27–40, 2000.
- [44] Albert L Herman and David B Spencer. Optimal, low-thrust earth-orbit transfers using higher-order collocation methods. *Journal of Guidance, Control, and Dynamics*, 25(1):40–47, 2002.
- [45] Kathryn F Graham and Anil V Rao. Minimum-time trajectory optimization of multiple revolution low-thrust earth-orbit transfers. *Journal of Spacecraft and Rockets*, 52(3):711–727, 2015.
- [46] Dong Eui Chang, David F Chichka, and Jerrold E Marsden. Lyapunov-based transfer between elliptic keplerian orbits. *Discrete & Continuous Dynamical Systems-B*, 2(1):57, 2002.
- [47] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1964.

- [48] Divya Garg, William W Hager, and Anil V Rao. Pseudospectral methods for solving infinite-horizon optimal control problems. *Automatica*, 47(4):829–837, 2011.
- [49] Lorenz T Biegler and Victor M Zavala. Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization. *Computers & Chemical Engineering*, 33(3):575–582, 2009.
- [50] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [51] William Karush. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.
- [52] Harold W Kuhn and Albert W Tucker. Nonlinear programming. In *Traces and emergence of nonlinear programming*, pages 247–258. Springer, 2014.
- [53] William Squire and George Trapp. Using complex variables to estimate derivatives of real functions. *SIAM review*, 40(1):110–112, 1998.
- [54] Jarrett Revels, Miles Lubin, and Theodore Papamarkou. Forward-mode automatic differentiation in julia. *arXiv preprint arXiv:1607.07892*, 2016.
- [55] David E Goldberg. Genetic algorithms in search. *Optimization, and Machine-Learning*, 1989.
- [56] Andrew R Conn, Nicholas IM Gould, and Philippe Toint. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.
- [57] Andrew Conn, Nick Gould, and Philippe Toint. A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Mathematics of Computation of the American Mathematical Society*, 66(217):261–288, 1997.
- [58] J. Shannon, M. Ozimek, J. Atchison, and C. Hartzell. Rapid design and exploration of high-fidelity low-thrust transfers to the moon. *Journal of Spacecraft and Rockets*, 2021. In Review.
- [59] J. Shannon, M. Ozimek, J. Atchison, and C. Hartzell. Rapid design and exploration of high-fidelity low-thrust transfers to the moon. In *2020 IEEE Aerospace Conference*, pages 1–12, 2020.
- [60] Craig A Kluever and Bion L Pierson. Optimal low-thrust three-dimensional earth-moon trajectories. *Journal of Guidance, Control, and Dynamics*, 18(4):830–837, 1995.

- [61] MT Ozimek and KC Howell. Low-thrust transfers in the earth-moon system, including applications to libration point orbits. *Journal of Guidance, Control, and Dynamics*, 33(2):533–549, 2010.
- [62] Albert L Herman and Bruce A Conway. Optimal, low-thrust, earth-moon orbit transfer. *Journal of Guidance, Control, and Dynamics*, 21(1):141–147, 1998.
- [63] Anastassios E Petropoulos, Zahi B Tarzi, Gregory Lantoine, Thierry Dargent, and Richard Epenoy. Techniques for designing many-revolution electric-propulsion trajectories. *Advances in the Astronautical Sciences*, 152(3):2367–2386, 2014.
- [64] Harry Holt, Roberto Armellin, Andrea Scorsoglio, and Roberto Furfaro. Low-thrust trajectory design using closed-loop feedback-driven control laws and state-dependent parameters. In *AIAA Scitech 2020 Forum*, 2020.
- [65] Bindu B Jagannatha, Jean-Baptiste H Bouvier, and Koki Ho. Preliminary design of low-energy, low-thrust transfers to halo orbits using feedback control. *Journal of Guidance, Control, and Dynamics*, 42(2):260–271, 2018.
- [66] Apollo fusion, inc., 2020. <https://www.apollofusion.com>, last accessed 2020-03-26.
- [67] Martin T Ozimek and Justin A Atchison. Nasa double asteroid redirection test (dart) low-thrust trajectory concept. In *27th AAS/AIAA Space Flight Mechanics Meeting, San Antonio, USA*, volume 221, 2017.
- [68] David Hadka and Patrick Reed. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary computation*, 21(2):231–259, 2013.
- [69] Christopher Rackauckas and Qing Nie. Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.
- [70] Virgin orbit, 2020. <https://www.virginorbit.com>, last accessed 2020-04-16.
- [71] Northrop grumman, 2020. <https://www.northropgrumman.com/space/pegasus-rocket/>, last accessed 2020-04-16.
- [72] J. Shannon, D. Ellison, and C. Hartzell. Exploration of low-thrust lunar swingby escape trajectories. In *AAS/AIAA Space Flight Mechanics Meeting. AAS Paper 21-273*, 2021.
- [73] Fernando Peralta and Steve Flanagan. Cassini interplanetary trajectory design. *Control Engineering Practice*, 3(11):1603–1610, 1995.
- [74] R Jehn, D Garcia, J Schoenmaekers, and V Companys. Trajectory design for bepicolombo based on navigation requirements. *Journal of Aerospace Engineering*, 4(1):1, 2012.

- [75] James V McAdams, David W Dunham, Robert W Farquhar, Anthony H Taylor, and BG Williams. Trajectory design and maneuver strategy for the messenger mission to mercury. *Journal of spacecraft and rockets*, 43(5):1054–1064, 2006.
- [76] David W Dunham, José J Guzmán, and Peter J Sharer. Stereo trajectory and maneuver design. *Johns Hopkins APL Technical Digest*, 28(1):104–125, 2009.
- [77] Daniel García Yárnoz, Chit Hong Yam, Stefano Campagnola, and Yasuhiro Kawakatsu. Extended tisserand-poincaré graph and multiple lunar swingby design with sun perturbation. In *Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques*, 2016.
- [78] Gregory Lantoine, Timothy P McElrath, and P Timothy. Families of solar-perturbed moon-to-moon transfers. In *24th AAS/AIAA Spaceflight Mechanics Meeting. AAS/AIAA*, 2014.
- [79] Timothy P McElrath, Gregory Lantoine, Damon Landau, Dan Grebow, Nathan Strange, Roby Wilson, and Jon Sims. Using gravity assists in the earth-moon system as a gateway to the solar system. 2012.
- [80] Jacob A Englander, Donald H Ellison, and Bruce A Conway. Global optimization of low-thrust, multiple-flyby trajectories at medium and medium-high fidelity. In *24th AAS/AIAA Space Flight Mechanics Meeting, 2014*, pages 1539–1558. Univelt Inc., 2014.
- [81] Jonathan Sims and S Flanagan. Preliminary design of low-thrust interplanetary missions. In *AAS/AIAA Astrodynamics Specialist Conference AAS Paper 99-338*, Girdwood, AK, 1999.
- [82] Nathan Strange, Damon Landau, Richard Hofer, J. Snyder, Thomas Randolph, Stefano Campagnola, James Szabo, and Bruce Pote. Solar electric propulsion gravity-assist tours for jupiter missions. In *AIAA/AAS Astrodynamics Specialist Conference*, 2012.
- [83] Jacob Englander, David Folta, Richard Hofer, and Sun Hur-Diaz. Optimization of the lunar icecube trajectory using stochastic global search and multi-point shooting. In *AIAA/AAS Astrodynamics Specialist Conference AAS Paper 20-435*, 2020.
- [84] CH Yam, DD Lorenzo, and D Izzo. Low-thrust trajectory design as a constrained global optimization problem. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(11):1243–1251, 2011.
- [85] Jacob A Englander and Arnold C Englander. Tuning monotonic basin hopping: improving the efficiency of stochastic search as applied to low-thrust trajectory optimization. 2014.

- [86] Hongru Chen, Yasuhiro Kawakatsu, and Toshiya Hanada. Earth escape from a sun-earth halo orbit using unstable manifold and lunar swingbys. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 59(5):269–277, 2016.
- [87] Jeremy Knittel, Kyle Hughes, Jacob Englander, and Bruno Sarli. Automated sensitivity analysis of interplanetary trajectories for optimal mission design. 2017.
- [88] Nathan J Strange and James M Longuski. Graphical method for gravity-assist trajectory design. *Journal of Spacecraft and Rockets*, 39(1):9–16, 2002.
- [89] J. Shannon, M. Ozimek, J. Atchison, and C. Hartzell. Analytical partial derivatives of the q-law guidance algorithm. *Journal of Guidance, Control, and Dynamics*, 2021. In Review.
- [90] J. Shannon, D. Ellison, and C. Hartzell. Analytical partial derivatives of the q-law guidance algorithm. In *AAS/AIAA Space Flight Mechanics Meeting. AAS Paper 21-274*, 2021.
- [91] Donald H Ellison, Bruce A Conway, Jacob A Englander, and Martin T Ozimek. Analytic gradient computation for bounded-impulse trajectory models using two-sided shooting. *Journal of Guidance, Control, and Dynamics*, 41(7):1449–1462, 2018.
- [92] Donald H. Ellison, Bruce A. Conway, Jacob A. Englander, and Martin T. Ozimek. Application and analysis of bounded-impulse trajectory models with analytic gradients. *Journal of Guidance, Control, and Dynamics*, 41(8):1700–1714, 2018.
- [93] Theodore N Edelbaum. Propulsion requirements for controllable satellites. *Ars Journal*, 31(8):1079–1089, 1961.
- [94] Etienne Pellegrini and Ryan P Russell. On the computation and accuracy of trajectory state transition matrices. *Journal of Guidance, Control, and Dynamics*, 39(11):2485–2499, 2016.